La Commode

Le Magazine des Ordinateurs Commodore

2e Année

n° 7
Avr. 83

VIC

64

CH. BLANCHER.

Le sommaire complet est en page 3

35 Francs

Sommaire

ELARGISSEZ

L'HORIZON DE VOTRE CBM 4000#
AVEC CP/MAKER

CP/Maker se présente sous forme d'une carte à installer à l'intérieur du CBM offrant les possibilités suivantes

. 64 K RAM supplémentaires accessibles en mode 6502

. Microprocesseur Z-80

. Système CP/M livré sur disquette format 4040 et 8050

- comportant CP/M , Assembleur , Editeur de texte , Utilitaire PIP (transferts entre périphériques)

Logiciels supplémentaires disponibles:

M/BASIC C/BASIC
BASIC Compiler CB 80
PASCAL MT-
CISCOBOL etc...



19-21, RUE MATHURIN-RÉGNIER - 75015 PARIS
TÉL. : 306.82.02 - TÉLEX : 204 875 F.

Valable également pour CBM 8032

EDITORIAL

Ce numéro est le 7ième de La Commode.

Depuis l'Antiquité le nombre 7 est considéré comme remarquable, comme marquant une étape. De là les sept jours de la semaine. Pour montrer les aptitudes du VIC à dessiner des jeux de caractères nous reproduisons ci-dessous, grâce à l'obligeance de M. Pascal DUCHIER de PROCEP, le paragraphe de la Genèse qui sanctifie le 7ième jour, en français et en hébreu . N'y voyez pas de propagande religieuse, mais une simple glorification du 7.

Le sept marque donc l'heure des bilans. Celui de La Commode est spécialement encourageant: lettres enthousiastes de lecteurs demandant une périodicité plus rapide (il y en a même un qui souhaite La Commode hebdomadaire), nombre d'abonnés en progression constante, renouvellement à 90% des abonnements terminés (au fait, et le vôtre?), tirages en progression. Bonne nouvelle pour nous, mauvaise pour certains: les numéros 1 à 4 sont totalement épuisés, mais rassurez-vous, nous publions l'album qui en réunit les articles de fond.

Enfin, le septième jour, c'est celui de la détente. Voyez notre double page "1er avril". Tout n'y est pas poisson d'avril.

Daniel-Jean DAVID

ויכלו השמים והארץ וכל
צבאם:
ויכל אלהים ביום
השביעי מלאכתו אשר עשה
וישבת ביום השביעי מכל
מלאכתו אשר עשה:
ויברך אלהים את יום
השביעי ויקדש אתו כי בו
שבת מכל מלאכתו אשר ברא
אלהים לעשות:

Ainsi furent achevés les Cieux et la Terre et tous leurs Hôtes.
Dieu termina son ouvrage le septième jour et Il se reposa en ce septième jour de tout le travail qu' Il avait accompli.
Et Dieu bénit ce septième jour et Il le sanctifia comme jour de repos (Sabbat) après tout son travail de Création.

Voici les gagnants du feedback des numéros 2 et 3:

no.2: Luc WALTERTHUM (57150 CREUTZWALD)
Alain GASNIER (69006 LYON)
no.3: Paul-André SAULOU (91450 SOISY)
Pascal NAVEL (75007 PARIS)

Communiqués de Presse

La société:

Jean Guipet
20, i rue de Fontaine Ecu
25000 BESANCON

commercialise un programme de métré bâtiment : "METRE B.E.G.".

*
* *

Les éditions:

BCM
24 route de la Sapinière
B-4960 BANNEUX
BELGIQUE

annoncent la sortie de leur nouvel ouvrage:

LE LIVRE DU VIC par B. MICHEL

Le prix de vente officiel est de 860 francs belges port compris (TVA comprise pour la Belgique).

B. MICHEL est également l'auteur de

PROGRAMMES INTERNES DU PET/CBM

également publié par BCM.

L'EPS (Ecole Professionnelle Supérieure) organise un stage conventionné par la Région Ile de France dans le cadre de la filière électronique:

Préparation à la fonction d'ingénieur électronicien

Ce stage est gratuit et rémunéré et s'adresse aux jeunes de plus de 18 ans, titulaires d'une maîtrise scientifique ou technique désirant se préparer aux fonctions d'Ingénieur électronicien.

Cette formation est programmée à partir du 16 mai 1983 (avec une interruption au mois d'août 1983). Il se déroulera à plein temps: 800 heures au centre EPS Yvelines et 400 heures en entreprise.

Les jeunes susceptibles de suivre cette formation de haut niveau doivent très rapidement faire acte de candidature soit auprès de EPS Yvelines, 25 rue Ambroise Croizat, 78280 GUYANCOURT (Melle TOUBLAN au (3) 043 57 90), soit auprès de EPS Paris, 45 rue des Petites Ecuries, 75010 PARIS (Melle PUYRAVAU au (1) 523 35 30).

Courrier des lecteurs

Je possède un VIC 20 :
1. Comment arrêter un programme pour utiliser le mode direct, puis reprendre ?

2. Il me semble que l'instruction LIST vide la mémoire des données ; qu'y faire ?

3. Peut-on trouver en France un traitement de texte pour le VIC?

4. Existe-t-il un équivalent de VISICALC pour le VIC ?

5. Je ne sais pas me servir de l'imprimante ; qu'y faire ?

GUY DELOFFRE
59219 FLOYON

1. Faire STOP. On reprend par CONT à condition que pendant votre mode direct :

a. vous n'ayez fait aucune erreur (message SYNTAX ERROR p.ex.)
b. vous n'ayez fait aucune modification au programme.

2. A notre connaissance LIST ne vide pas la mémoire des données.

3. Il devrait être disponible chez PROCEP à la date de publication de ce numéro.

4. SIMPLI-CALC devrait être disponible lui aussi chez PROCEP.

5. Suivez les articles de La Commode qui en parleront. Voir aussi D.J. DAVID "La Pratique du VIC" vol I, ed. du PSI.

*
* *

1. J'ai essayé de reloger le VICMON en RAM, mais ne suis pas parvenu à le faire tourner car le VIC se plante lorsque l'on veut modifier certaines adresses ($0314-$0319).

2. J'aimerais bien que paraisse "LA PRATIQUE DU VIC 20 - Langage machine" afin de me faciliter l'étude des programmes en assembleur. Son édition est-elle envisagée et pour quand ?

Paul BOURDONNEAU
26100 ROMANS

1. La façon de reloger VICMON est donnée dans VIC à BRAC du prochain numéro. C'est effectivement une opération délicate.

2. "La Pratique du VIC vol II" est en préparation, ainsi que "Clefs pour le VIC". Vous serez informé de leur parution par les annonces de PSI.

*
* *

Dans le numéro 5 de la Commode page 43, il y a un programme sur le loto :

1. Le programme est-il adapté pour le VIC 20?

2. Sinon, est-il possible de se procurer des explications plus détaillées de ce programme afin de pouvoir l'étudier?

Après plusieurs essais, apparaissaient toujours les mêmes erreurs : lignes qui se sont introduites et qu'il est impossible d'effacer.

Pour ces raisons, j'ai fait

vérifier mon VIC, et les mêmes erreurs sont apparues.

André MARMOUGET
92700 COLOMBES

1. Ce programme n'est pas fait pour le VIC.

2. Pour l'adapter au VIC les points à regarder sont :

- largeur de l'affichage
- ligne 210 : dépend de l'adresse d'écran
- lignes 360 à 390 : obtention d'un "bip". Se fait autrement sur VIC (POKE 36876...)
- lignes 540 à 620 : dépendent des paramètres d'écran : 40 doit devenir 22.
- ligne 710 : arrête le son (POKE 36878,0)
- ligne 720 : envoie le curseur en un point de l'écran.
- ligne 730 : son.

*
* *

1. Pouvez-vous m'indiquer comment passer un programme écrit pour le VIC 20 de base, alors que j'ai maintenant l'extension 8 K ?

En particulier pour le programme "BLITZ" vendu par PROCEP.

2. Existe-t-il un répertoire des adresses mémoire permettant une approche plus précise des POKE et PEEK ?

3. Je souhaiterais acquérir un ouvrage plus complet sur le VIC, j'ai actuellement :

- la notice du constructeur
- le livre de D-J. DAVID "La Découverte du VIC".

4. Existe-t-il une notice concernant le fonctionnement des imprimantes ? Si oui, comment se la procurer ?

Jacques PIGNOT
90800 BAVILLERS

1. Le sujet a commencé à être traité dans La Commode numéro 6 et continue à l'ètre dans ce numéro.

En bref : il faut paramétrer les adresses d'écran (voir La Commode numéro 3 page 31) et faire commencer BASIC en $2000 par POKE 44,32 : POKE 46,32 : POKE 48,32:CLR

2. La Commode donnera bientôt des modes d'emploi précis de ces adresses.

3. Vont bientôt paraître chez PSI :

"La Pratique du VIC" vol I de D-J- DAVID.
"Clefs pour le VIC" de D-J-DAVID.

Chez PROCEP, il y a en anglais, mais bientôt en français :

"VIC Revealed", de Nick Hampshire
"VIC Programmer's reference guide".

4. Normalement l'imprimante du VIC est accompagnée d'une notice en français. Exigez-la de votre distributeur. D'autre part, il y aura des articles dans La Commode.

*
* *

La COMMODE ? Joli meuble pour l'esprit, ma foi :
sérieux sans se prendre au sérieux et à mi-chemin entre le jargonnant esotérique et la pseudo-vulgarisation bêtifiante. D'ailleurs, je dois à votre estimable publication de fructueux échanges avec un de vos lecteurs (je vous salue, lecteur !) et ce n'est qu'un début.

Alors, hop ! je m'abonne -voir chèque- (harmonieux tintement du tiroir-caisse).

Et puis, tiens, deux suggestions :

1. Certaines revues américaines, lorsqu'elles publient un programme pour PET/CBM ajoutent systématiquement toutes les modifications à apporter au programme de base selon qu'on est VICieux, HUIT-MILLiant, QUATRE-MILListe petit ou grand écran, TROIS-MILLien ou PETomane de

la première heure. La COMMODE l'a déjà fait. Ne pourrait-on pas devenir un parti pris systématique- et ô combien courtois ?

2. Deuxième couplet : après pas mal de cafouillages, mon LIBRARIAN tourne enfin (merci aux chevronnés qui m'ont conseillé) ! A moi les joies de l'APPEND et du MERGE, que je me disais. Déception et Lacrymosa ! LIBRARIAN tourne sur fichiers séquentiels et constitue donc un système à part. Daniel-Jean DAVID, notre père à tous, ne pourrait il pas nous mitonner un petit programme ADLIB permettant de transformer un programme déjà sur disque (PRG) en fichier SEQ digestible par LIBRARIAN ? Allez, DJD, c'est oui, hein ? Promis ? Dites ? Oh ! Siou plait et tout ça! C'est dit ? Oui ? Alors un grand merci sur 132 colonnes et continuez, vous êtes dans le bon sens.

François J. BAYARD
53000 LAVAL

1. Nous avons commencé dans le numéro 6 à préciser toutes ces modifications à la fin de l'article. Cela devrait progressivement figurer dans la plupart des articles.

2. D'abord chargez le programme en question, il doit n'avoir qu'une instruction par ligne. Ensuite faites :

```
OPEN 8,8,8,"0:ZOZO,S,W"
CMD 8 : LIST
```

*
* *

1. J'ai eu quelques difficultés dans la PRATIQUE DU PET/CBM vol.I
. Page 78 FICHIER DIRECT CREATION SEQUENTIELLE.
le print 84 de la ligne 4160 ne fonctionnait pas, le nombre de blocs écrits restait obstinément à 0. J'ai fait passer la ligne 4025 en 4155, suivi de 4156 print 84,nr$ cr$;(ligne 4030), puis 4027 la ligne 4055.

Comme cela, ça fonctionne sur ma 2031 .

2. La commande COPY de la dite 2031 me pose quelques problèmes. En tenant compte des recommandations de J. PIERRAT numéro 4 page 59, j'ai réussi à "concaténer" deux fichiers séquentiels (il suffit de supprimer l'index de fin du premier fichier pour pouvoir lire le nouveau fichier créé en entier.

Par contre, pour les programmes, le nombre de blocs m'indique que les deux fichiers sont bien rassemblés, mais après chargement le listing s'arrête au premier ?

Jacques BOUTIQUE
87000 LIMOGES

1. Merci des rectifications pour 2031. Nous allons essayer de creuser le problème.

2. Le problème que vous posez n'en est pas un : les fichiers programmes ne peuvent se concaténer par cette méthode car la commande ne fait que mettre les fichiers bout à bout sans établir le chaînage entre instructions.

*
* *

1. Est-il possible de trouver un générateur de caractères, permettant d'obtenir les caractères accentués sur la 4022 et ne nécessitant pas de programme de transcodage (ce qui est le cas de la ROM commercialisée par BCM, qui n'utilise pas les mêmes codes que PROCEP pour ses caractères accentués) ?

Si un tel produit existe, qui le commercialise ?

2. Quelles modifications peut-on effectuer sur la 4022, pour imprimer en haute résolution, c'est-a-dire pour faire une impression point par point. La méthode utilisant le caractère programmable est très longue à utiliser.

3. Je possède d'autre part le logiciel Pascal pour système 8001 vendu par PROCEP. Or la procédure VDU est incorrecte. Elle considère en effet que l'écran n'a que 40 colonnes. Comment peut-on modifier cette procédure ?

Jean-Claude VILLE
97438 SAINTE MARIE

1. A notre connaissance, il n'existe pas d'autre générateur de caractères pour 4022 que celui de BCM. Il serait peut-être intéressant d'en produire un sur EPROM...

2. Seule la méthode du caractère programmable est applicable. Sinon, on peut constituer sur EPROM un générateur de caractères ayant pour "caractères" tous les segments possibles traversant une maille: c'est donc adapté au tracé de courbes.

3. PROCEP possède maintenant une version corrigée de Pascal 8000. S'adresser à Ph. HAGEGE de PROCEP.

*
* *

J'ai acquis le VIC 20 il y a 6 mois ; j'ai beaucoup hésité avec son concurrent direct le ZX81, et j'avoue en être très satisfait. Le seul inconvénient est qu'il n'y a que 23 lignes et 22 colonnes.

1. Le branchement VIC-TV donne la même image couleur, que ce soit avec le câble d'antenne ou avec la prise Péritel, pourquoi ? quel est alors l'avantage de la péritel ?

2. Après avoir "loadé" un programme, comment réaliser un autorun ?

3. Lorsqu'on fait REM suivi de caractères graphiques, par exemple 10 REM ??????? on tape ensuite un RUN, il ne se passe rien, mais si on LISTe, alors on obtient:
10 REM INPUT IF VAL. Peut-on utiliser cette particularité de la fonction REM dans un programme?

4. Je suis très satisfait du livre "La Découverte du VIC". Il est bien expliqué et complet, sauf pour la fonction ON. Après m'être procuré le VICMON, j'ai acheté "Pratique du PET/CBM" tome II, qui m'a beaucoup déçu ; c'est pourquoi je vous demande si possible un peu plus de programmes en assembleur en donnant pour chaque micro-ordinateur les adresses correspondantes, car j'estime qu'il faut profiter du fait que l'on a tous le 6502.

Ramdane FERHATI
59000 LILLE

1. Normalement, le branchement Péritel donne une meilleure image, mais si le câble d'antenne est satisfaisant, on ne voit pas la différence.

2. L'autorun sera traité prochainement dans La Commode.

3. La particularité du REM vient de ce que les codes des caractères graphiques sont égaux aux codes des mots clés basic. Mais si vous mettez les caractères entre guillemets ("..."), vous n'aurez pas les mots clés au LIST.

4. "La pratique du PET" vol II n'est pas adapté au VIC sauf pour la description du jeu d'instructions du 6502. Les adresses stratégiques, en particulier sont à changer ; vous pouvez le faire grâce aux tables des numéros 3 et 4 de la Commode.

*
* *

Magazine

Le SK c'est exquis

Dès maintenant disponible, un nouveau modèle, le CBM 8032 SK / 8096 SK (SK = Separate Keyboard, clavier détachable).



La photo ci-dessus montre qu'il s'agit d'une caisse de CBM700 avec l'électronique du CBM 8000: le clavier reste celui du 8000 et il n'y a pas , sous l'écran, les disquettes qu'aura le 700. A propos, cette sortie ne laisserait-elle pas supposer un retard de la sortie du 700?

La bourse ou la vie!

Excellent résultat pour les actions de Commodore aux Etats-Unis pour 1982 où on a observé le classement:

1	Commodore	+103,7%
2	Tandy	+49,8%
3	Apple	+35,0%
4	Zenith	+29,2%

Rappelons qu'il ne s'agit pas des ventes (voir ci-dessous) mais des cours des actions. On observe qu'Apple ne fait que suivre l'augmentation moyenne sur la période (+32,8%).

Les ventes du VIC

Il y a maintenant plus d'un million de VIC dans le monde, dont 800 000 aux USA.

En France, les ventes du dernier trimestre 82 ont approché les 8000 unités.

64 - 64 - 64

On espère la commercialisation effective du Commodore 64 dans le courant du 2ème trimestre 83. Il est déjà présent en 1 exemplaire chez chaque distributeur pour familiarisation et démonstration.

Vous avez dit CP M ?

Gros efforts pour adapter CP/M aux divers matériels Commodore. Une version de CP/Maker est disponible pour CBM4000 grand ou petit écran. Elle fera l'objet d'une offre spéciale PROCEP.

Une option CP/M est d'autre part annoncée pour le 64.

Logiciels 64: take it easy

Pendant que nous en sommes au 64, parlons des premiers logiciels annoncés. De même que les logiciels du VIC s'appellent Simpli-xxx, ceux du 64 s'appelleront Easy-xxx.

Easy-calc: version 64 du célèbre VisiCalc. C'est un logiciel de

traitement de tableaux de nombres à cases liées par des relations: ce que la récente norme demande d'appeler un tableur.

Easy-plot: logiciel de production de tableaux et de graphiques.

Easy-finance: analyse de prêts ou de placement, valeurs actualisées.

Easy-schedule: logiciel de suivi de projets dans le temps.

Easy-script: traitement de texte perfectionné pour un prix modique.

et aussi, bien sûr, des jeux et des programmes d'enseignement sur lesquels un gros effort de traduction en français est en cours. A ce propos, PROCEP recrute des collaborateurs à domicile (prêt d'un 64 à la clé) pour cette tâche, voir en page "petites annonces".

En outre Commodore a conclu un accord avec les créateurs du langage LOGO pour son adaptation au 64. Rappelons que FORTH sera aussi disponible.

En Suisse

Commodore est en tête de la micro-informatique Suisse avec un parc de 7500 CBM et 2500 VIC.

Le 64 est d'ores et déjà commercialisé (depuis décembre) comme le montre l'article de ce numéro qui nous a été envoyé par un lecteur suisse.

Le voile se lève ...

... sur les nouveaux composants MOS Technology utilisés dans le 64.

Le microprocesseur 6510 semble bien être exactement un 6502 mais possédant (aux adresses 0, registre direction et 1, registre données) un port de 8 bits parallèles. Le brochage est aménagé en conséquence. Ce port sert à commander l'activation de certaines zones mémoire: on peut désactiver complètement la mémoire morte et avoir l'espace complet de 64K consacré à la mémoire vive.

Attention aux espions de la CIA

Le nouveau boitier 6526 ressemble beaucoup au 6522. MOS Technology, ou plutôt "Commodore Semiconductor Group" l'appelle CIA (Complex Interface Adaptor) mais il n'est pas à notre avis plus complexe que le 6522. Il n'a pas les broches de "handshaking" des ports parallèles mais a des entrées-sorties des temporisateurs rendant plus clair leur fonctionnement en compteur. Un troisième temporisateur sert d'horloge temps réel. Comme le 6522, il a 16 registres internes.

COMMODOSBORNE

A la récente Winter Consumer Electronics Show de Las Vegas, Commodore a montré un prototype appelé SX100 constitué d'un coffret contenant un moniteur couleur et deux platines disquettes, ayant pour couvercle un clavier et bien sûr, une poignée de transport. En somme, la structure de l'Osborne 1, mais l'électronique du 64!

A cette même exposition Commodore a montré:

- un hand-held computer style Sharp PC 1500,
- un VIC muni d'un écran télé miniature incorporé,
- un clavier style orgue pour exploiter les possibilités du 64 en synthèse musicale,
- un synthétiseur de voix. Il faut noter que Commodore vient de créer une nouvelle division, la "Speech Technology Division" basée à Dallas (rien à voir avec JR...).

Bientôt votre VIC et votre 64 parleront.

Nouveau magnéto-cassette

Aperçu en photo un nouveau modèle de magnétocassette plus petit que celui que nous connaissons. Au point de vue élégance il est à peu près ce qu'est une 1540 par rapport à la 2031.

Pierre-Etienne THALBERG

Nouveautés VIC

De nouveaux programmes pour VIC 20 ont vu le jour au cours du premier trimestre 83. Ce sont SIMPLICALC, VICWRITER, VICFILE, WORDCRAFT 20 , QUIZMASTER, et des programmes d'aventure de Scott Adams.

SIMPLICALC

SIMPLICALC est une feuille de calcul électronique pour le VIC 20. L'utilisation de ce programme qui nécessite VIC 20 avec extension 16k, VIC 1540 (unité de disque) ou VIC 1530 (lecteur de cassettes) et VIC 1515 (imprimante), peut se faire aussi bien pour des analyses de bilan, que pour un budget familial ou une estimation de réparations.

VICWRITER

VICWRITER est un programme de traitement de texte qui vous permet de créer, sauvegarder et modifier facilement un texte sur un système composé d'un VIC 20 et d'une extension de 8 ou 16k, d'une unité de disque (VIC 1540) ou d'un lecteur de cassettes (VIC 1530) et d'une imprimante (VIC 1515).

L'utilisation de VICWRITER est assimilable à celle d'une machine à écrire. Les caractères apparaissent sur une ligne de la zone basse de l'écran, avec retour à la marge en fin de ligne. Une simple commande permet de choisir la largeur des marges et de créer un texte composé de lignes de 20 à 80 caractères.

VICFILE

VICFILE est le premier programme de gestion de fichiers conçu pour le micro-ordinateur VIC 20. Il nécessite un VIC 20 avec extension 16k, unité monodisquette (VIC 1540) et une imprimante (VIC 1515). Il offre de nombreuses caractéristiques présentées habituellement par les gestionnaires de fichiers d'un coût très élevé.

La particularité essentielle de VICFILE réside dans sa facilité d'adaptation à une grande variété d'applications d'affaires ou familiales. L'utilisateur peut mettre en oeuvre diverses options lui permettant de choisir la présentation des documents imprimés ou affichés à l'écran.

VICFILE fonctionne sur unité de disquettes grâce à laquelle vous mémoriserez jusqu'à mille fiches par disquette. En fait, cette quantité est fonction du nombre de rubriques contenues dans les fiche; ces dernières sont stockées dans un fichier en accès direct.

VICFILE offre également la possibilité de trier les fiches par ordre alphabétique ou numérique à partir de toute rubrique contenue dans les fiches. De plus, il est possible d'extraire un fichier ou de sélectionner des fiches selon des critères pré-définis, puis de les afficher à l'écran, de les imprimer soit séparement, soit sous forme d'état.

WORDCRAFT 20

Ce programme de traitement de texte se présente sous la forme d'une cartouche comprenant à la fois le programme et une extension mémoire de 8k octets qui sert de zone de travail à WORDCRAFT 20, mais aussi de mémoire vive supplémentaire lorsque l'on se trouve sous contrôle Basic. Il permet de stocker les textes créés aussi bien sur cassette que sur disquette. Un petit programme Basic fourni permet de relire la cassette contenant le texte créé sur VIC 20 sur un CBM 8000, et de transformer le texte pour l'utiliser avec TRAITEXT.

QUIZMASTER

QUIZMASTER est un programme sur cassette permettant la création et la mise en oeuvre de jeux de questions-réponses. Ce programme, qui fonctionne sur un VIC 20 avec extension de 8k ou 16k, est destiné à une utilisation familiale et associe le goût de la compétition et du jeu à une valeur éducative certaine. Les différentes options de ce programme sont : création, modification, sauvegarde et chargement à partir de la cassette, et utilisation du questionnaire. Les réponses sont de trois types: vrai ou faux, réponse exacte ou choix mul-

tiple. Enseignants et parents trouveront QUIZMASTER très enrichissant pour aider les enfants dans leurs exercices de révision.

* *
*

Les programmes d'aventure de Scott Adams constituent une nouvelle catégorie de jeux mettant en oeuvre un scénario où vos facultés d'imagination et de déduction seront mises à l'épreuve (comme pour un détective chargé de résoudre une énigme policière). Les scénarios d'aventure sont très variés et passionnants et ont déjà conquis les Etats-Unis et l'Angleterre.

Ces programmes d'aventure de Scott Adams, en anglais, sont sous forme de cartouche. Le manuel d'utilisation est en francais.

VIC 1914 ADVENTURELAND

Au pays de l'AVENTURE (Adventureland), vous devez ramasser une série de treize (13) trésors (Scott Adams n'est pas superstitieux, et vous ?) et les mettre en sécurité dans une chambre forte. La plupart des programmes d'AVENTURE se situent sous terre, avec quelques brèves apparitions au sol pour vous donner air frais et lumière du jour.

Il faudra prendre garde aux Dragons, Abeilles, Chauve-souris, Ours.

VIC 1915 PIRATE COVE

Dans PIRATE COVE, vous dèvez d'abord trouver où se situe l'Ile aux Trésors depuis votre appartement, et ensuite chercher ces trésors qui ont été enterrés quelque part dans l'île. Des incantations magiques pourront vous aider, mais la "charpenterie" est un art essentiel que vous apprendrez au cours de cette AVENTURE. Une certaine connaissance du comportement animal pourra être utilisée, mais pourra aussi être une épreuve fallacieuse.

Pirates ivres, Perroquets, Galériens apparaissent dans cette AVENTURE. Bien que les trésors ne soient pas tout à fait ce que vous vous attendriez à trouver pour un pirate, ils sont tous deux de valeur, chacun dans leur domaine.

VIC 1916 MISSION IMPOSSIBLE

Vous devez exercer votre intelligence contre un saboteur, dans une véritable course contre la montre. Il faut désamorcer à temps une bombe dans une Centrale Nucléaire. Vous devez trouver tous les laisser-passer permettant d'explorer toutes les parties de la Centrale. Ici il n'y a pas de trésor à trouver, mais seulement quelques objets indispensables à votre mission.

Toutes les informations techniques nécessaires peuvent être trouvées à l'intérieur de la Centrale Nucléaire. Aucune connaissance de physique nucléaire ou de déminage n'est nécessaire. Il est cependant recommandé d'avoir participé à d'autres programmes d'AVENTURE avant d'attaquer celui-là.

VIC 1918 VOODOO CASTLE

Vous avez été pris au piège à l'intérieur de l'enceinte du château de DRACULA, et vous devez vous débarrasser du Comte avant qu'il ne vous tue. Vous pouvez errer autour du château à volonté jusqu'à la tombée de la nuit. Puis vous commencez à vous sentir fatigué et vous vous endormez à moins que ...

Tous les instruments nécessaires à la destruction du Comte et à votre survie peuvent être trouvés à l'intérieur de l'enceinte du château, ainsi que quelques faux indices, et des détails sans intérêt.

VIC 1917 THE COUNT

Vous devez sauver le Comte Christie de l'anathème avant que vous ne soyez tué ou maudit vous-même. Magiciens et sorciers abondent à l'intérieur du château. Faites quelques pas dans le cimetière et observez les macabres avertissements. Prenez garde à la sorcière, mais rappelez-vous que tout n'est pas contre vous (seulement la plupart des choses!).

Marc VAN CRAEYNEST

La Course des compilateurs BASIC

Enfin, un des articles les plus attendus de La Commode.

Il existe deux compilateurs BASIC sur CBM: DTL et PETSPEED. Cet article essaie de vous indiquer lequel choisir compte tenu de vos applications. On analyse également le TINY BASIC COMPILER d'Abacus Software qui est aussi disponible sur le VIC.

Pourquoi un compilateur BASIC ?

Tout programme en langage évolué a besoin d'être traduit en langage machine pour être exécutable. Cette traduction est effectuée par un interpréteur ou un compilateur.

Avec un interpréteur, le texte source (votre programme) et l'interpréteur coexistent en mémoire (votre programme étant en mémoire vive-MEV, et l'interpréteur étant en mémoire morte-MEM). L'interpréteur considère successivement chacune de vos instructions: il la traduit puis l'exécute, puis passe à l'instruction suivante.

Un compilateur au contraire traduit tout votre programme en bloc, puis le code objet (= le programme traduit) est exécuté en bloc. Il y a en fait trois phases:

1. édition pour constituer ou corriger le texte BASIC et le stocker sur disque
2. la compilation proprement dite: la mémoire est occupée par le compilateur, le code objet va sur disque.
3. chargement du code objet en mémoire à la place du compilateur et exécution.

Un compilateur est plus avantageux qu'un interpréteur du point de vue de l'économie de mémoire et de la rapidité. Le programme compilé prend moins de place que le programme-source (entre autres tous les REM sont éliminés) et son exécution est plus rapide: dans chaque expression où une variable intervient, le compilateur place l'adresse de cette variable, on l'a donc tout de suite à l'exécution alors qu'avec l'interpréteur, l'adresse est recherchée dans la table des variables lors de chaque exécution des instructions où intervient cette variable. Autre élément parmi d'autres: dans le cas d'une boucle, les instructions de la boucle sont traduites une fois pour toutes à la compilation; avec un interpréteur chaque instruction de la boucle est traduite à chaque itération.

Ces avantages des compilateurs sont contrebalancés par des avantages des interpréteurs. Le processus de correction d'une erreur est bien moins lourd avec un interpréteur: le programme source est toujours en mémoire, il suffit de lister la ligne fausse et de la corriger par mouvements de curseurs. On peut alors immédiatement exécuter la nouvelle version. Avec un compilateur, il faut recharger le programme-source, le corriger, le restocker puis le recompiler, donc recharger le compilateur. Enfin charger le nouveau code objet et exécuter.

En cas d'erreur, avec un compilateur, on ne peut avoir comme indication que, par exemple, "division par 0 à tel endoit du code objet". La correspondance avec l'endroit précis du programme-source n'est pas évidente. Au contraire, l'interpréteur n'exécutant qu'une instruction-source à la fois est capable de dire où dans le programme-source il a trouvé une erreur.

Tout ceci montre qu'un interpréteur est plus avantageux pour mettre au point un programme mais qu'un compilateur est plus efficace

s'il s'agit d'exécutions répétées d'un programme au point (il n'y a plus à recompiler: on ne fait qu'exécuter le programme objet). Il en résulte que la situation idéale consisterait à disposer d'une part d'un interpréteur pour mettre au point le programme, et d'utiliser un compilateur pour exécuter le programme définitif.

C'est le cas sur le CBM où on dispose de l'interpréteur bien connu, et de deux compilateurs.

Critères de choix

Puisqu'on a deux compilateurs, il va falloir les comparer. Sur quels critères?

Tout d'abord le prix. A qualité égale cela joue.

Ensuite les bénéfices apportés donc principalement le facteur de gain en vitesse obtenu.

La facilité de mise en oeuvre est importante aussi. On a vu que l'utilisation d'un compilateur est lourde par rapport à un interpréteur. Tout ce qui la facilite est donc important.

Mais le critère le plus important est la compatibilité, c'est à dire dans quelle mesure le compilateur accepte un texte-source identique à celui qu'on a mis au point avec l'interpréteur. Une parfaite compatibilité est nécessaire si l'on veut profiter à plein des avantages de disposer à la fois d'un interpréteur et d'un compilateur.

Bien sûr, la compatibilité ne peut être totale. Si votre programme se manipule lui-même par des POKE en mémoire programme, il ne pourra être compilé puisque dans le cas de la compilation, le texte source n'est plus en mémoire lors de l'exécution.

De même certaines astuces de page zéro ne pourront se compiler puisque le compilateur peut avoir sa propre gestion de la page zéro.

Enfin aucun dimensionnement dynamique des matrices n'est possible: le compilateur doit avoir des constantes pour faire les réservations mémoire.

Sur d'autres points, il peut y avoir des semi-incompatibilités: les programmes qui font des délais avec des boucles doivent être modifiés car, avec un compilateur, les boucles se déroulent plus vite.

Un autre élément de compatibilité est de savoir si l'on a compatibilité seulement avec BASIC seul ou aussi avec EDEX, MASTER etc...

Comparaison des compilateurs DTL et PETSPEED

Ces deux compilateurs se ressemblent assez par leur prix: 3463F TTC pour DTL, 3750F TTC pour PETSPEED, leur présentation, une disquette et une clé électronique. Chacun a une notice adéquate, en anglais pour DTL, un peu moins détaillée mais en français pour PETSPEED. PETSPEED nous a été prêté pour cet essai par la société ELLIX que nous remercions.

Le mode d'emploi est assez analogue pour les deux compilateurs: le compilateur est sur la disquette 0, le texte-source est sur la disquette 1.

Le texte source a la même forme pour les deux compilateurs, soit tout simplement le fichier programme obtenu par la commande DSAVE de l'interpréteur BASIC. Pour modifier ce fichier-source vous utilisez les possibilités d'édition de l'interpréteur BASIC.

Le texte objet forme lui aussi un fichier-programme sur la disqette 1. Ce fichier sera chargé par DLOAD (puis RUN) à chaque fois que vous voudrez exécuter le programme compilé.

PETSPEED ne demande que le nom du fichier-source. Il construit à partir de là les noms des fichiers dont il a besoin. Notamment, si le fichier-source s'appelle ZOZO, le fichier objet s'appellera ZOZO.GT.

DTL pose une série de questions: nom du fichier source, nom du fichier objet (le programmeur a la liberté de spécifier le nom qu'il veut, sinon, par défaut DTL a des procédures d'assignation automatique du nom objet: si le fichier source s'appelle TOTO.SRC, le fichier objet s'appellera TOTO), option d'impression du listing sur imprimante. La fig. 1 est un exemple de ce qu'on peut obtenir pour un très court programme.

Ces possibilités de listing

```
********************************************************************
                       DTL-BASIC COMPILER  . (3.4.1)

SOURCE :CALESS   OBJECT :COMCALESS   ERRORS :Y   STATS :Y   RUN :X
********************************************************************


 10 TI$="000000"
 20 FOR I=1 TO 100000
 30 NEXT
 40 PRINTTI$



ERRORS -  0            WARNINGS -  0


LIBRARY ROUTINE SIZE -  4144( 16 BLOCKS, 48 BYTES)
DATA STATEMENT SIZE  -     0( 0 BLOCKS, 0 BYTES)
COMPILED CODE SIZE   -    38( 0 BLOCKS, 38 BYTES)
VARIABLE LIST SIZE   -     7( 0 BLOCKS, 7 BYTES)
ARRAY TABLE SIZE     -     0( 0 BLOCKS, 0 BYTES)

TOTAL PROGRAM SIZE   -  4182( 16 BLOCKS, 86 BYTES)
TOTAL FILE SIZE      -  4189( 16 BLOCKS, 93 BYTES)
```

Fig.1 - Exemple de listing obtenu sur imprimante avec DTL.

sur imprimante sont absentes de PETSPEED.

Pour l'exécution du programme qu'on vient de compiler, PETSPEED offre une facilité supplémentaire: il est inutile de charger; le programme objet est resté en mémoire. Il suffit de faire "Home" (le curseur va sur un RUN) et "Return".

La comparaison la plus importante est celle de la vitesse. Pour cela nous avons développé deux programmes de benchmark, avec quelques variantes supplémentaires. Le premier (fig. 2) est un produit de matrices classique, en dimension 20x20, 30x30 et 40x40. Dans une variante de MATRICE 20x20 (fig. 3), on a ajouté des REM et des variables dimensionnées avant les matrices à multiplier: cela allonge les recherches dans le cas interprété.

Le tableau 1 résume les résultats obtenus. La phase 1 est l'initialisation de 5 à 45, la phase 2 est le produit proprement dit.

On a indiqué aussi dans le tableau les facteurs de gain (sous la forme nx).

On voit deux choses:

1- les résultats sont très décevants par rapport aux facteurs annoncés dans la littérature. Ceci est dû en fait à ce que notre programme est écrit de façon performante (pas de REM, pas de GOTO). On voit d'ailleurs que les facteurs de gain s'améliorent si l'on prend la version "allongée", alors que les temps des versions compilées sont rigoureusement constants. On aura des facteurs bien meilleurs avec le 2ième Benchmark.

2- PETSPEED est spectaculairement plus rapide que DTL, d'un facteur 2.5 environ.

Il faut noter deux incidents qui ont marqué l'essai de MATRICE 40x40.

- PETSPEED a refusé le programme par manque de mémoire. C'est donc que son interpréteur de code compilé est assez encombrant (8K), plus que celui de DTL, mais il est vrai qu'il est plus rapide. Maintenant, c'est juste quand on a de grosses matrices à multiplier qu'on a besoin de la rapidité d'un compilateur or voilà qu'il nous lâche: il aurait en fait été jusqu'à 36x36.

- DTL a exécuté correctement le produit, mais après, la moitié

```
5 PRINTTI$
10 N=20:DIMA(20,20),B(20,20),C(20,20)
20 FOR I=1 TO N :FOR J=1 TO N
30 A(I,J)=I :B(I,J)=J
40 NEXT J :NEXT I
45 PRINTTI$
50 FOR I=1 TO N :FOR J=1 TO N
60 C(I,J)=0
70 FOR K=1 TO N
80 C(I,J)=C(I,J)+A(I,K)*B(K,J)
90 NEXT K :NEXT J :NEXT I
100 PRINTC(10,10)
110 PRINTTI$
```

Fig.2 - MATRICE 20x20.

```
5 PRINTTI$
7 DIMS(5),T(5),U(5),V(5),W(5),X(5),Y(5),Z(5)
10 N=20:DIMA(20,20),B(20,20),C(20,20)
20 FOR I=1 TO N :FOR J=1 TO N
25 REM  INITIALISATION
30 A(I,J)=I :B(I,J)=J
40 NEXT J :NEXT I
45 PRINTTI$
50 FOR I=1 TO N :FOR J=1 TO N
60 C(I,J)=0 :REM MISE A ZERO
70 FOR K=1 TO N
75 REM  BOUCLE INTERNE
80 C(I,J)=C(I,J)+A(I,K)*B(K,J)
90 NEXT K :NEXT J :NEXT I
100 PRINTC(10,10)
110 PRINTTI$
```

Fig.3 - MATRICE 20x20 "allongé".

VERSION	INTERPRETEUR	DTL	PETSPEED
20x20 phase 1	6 s	4 s 1,5x	1 s 6x
20x20 phase 2	195 s	118 s 1,65x	48 s 4x
30x30 phase 1	13 s	7 s 1,85x	20 s 6,5x
30x30 phase 2	10mn 52 = 652 s	398 s 1,64x	162 s 4x
40x40 phase 1	22 s	12 s 1,83x	ne passe pas
40x40 phase 2	25mn 34 = 1534 s	934 s 1,64x	
20x20 allongé	7 s (phase 1)	4 s 1,75x	1 s 7x
20x20 allongé	218 s (phase 2)	118 s 1,84x	48 s 4,54x

Tableau 1.

de la disquette 1 s est trouvée scratchée sans que nous ayons idée de ce qui s est passé réellement (probablement un fichier intermédiaire trop encombrant?). C est gênant. Moralité ayez des backup!

Le second programme de benchmark (fig. 4) fait lui tout ce qu il ne faut pas faire avec un interpreteur: utiliser une constante dans une boucle (gérée directement et non par FOR) et avoir des REM en début de programme qui allongent la recherche de l instruction 50 lors du THEN 50 de la ligne 60.

```
5 PRINTTI$
10 REM CALCUL DE PI
20 REM ON ASSIGNE 10000 FOIS
30 REM LA MEME VALEUR
40 I=1
50 PI=3.14159265
60 I=I+1:IF I<10000 THEN 50
70 PRINTTI$,PI
```

Fig.4 - Programme PI.

Résultat en interprété 7mn 56 soit 476s avec DTL 39s soit 12x et avec PETSPEED 11s soit 43x. Mais notez bien qu on n a pas toujours ces gains. On a des gains importants aussi avec l intruction GET notammment lorsqu on construit une chaîne par concaténation caractère par caractere. En revanche il n'y a pratiquement pas de gain sur les PRINT.

Compatibilité et autres critères

Tant PETSPEED que DTL se sont montrés parfaitement compatibles avec BASIC interprété (sous réserve de tenir compte des remarques du début de cet article).

L'un et l'autre ont quelques extensions à BASIC sous forme d'extensions de l'interprétation de certaines instructions BASIC.

DTL et PETSPEED ont une meilleure gestion des entiers et des variables entières. Ils admettent les variables % dans FOR

PETSPEED a en plus deux extensions très intéressantes:

+ variables longues (avec plus que les deux premiers caractères significatifs) obtenues grâce à la directive REM!LN

+ DEF FN peut définir une fonction chaîne de caractères ou à argument chaîne:

Ex DEF FN(B$)=LEFT$(B$+" " 10)

Ceci est très intéressant mais contradictoire avec notre critère de compatibilité.

EDEX

Un autre élément de compatibilité très important qui peut pour certaines applications contrebalancer le critère de vitesse est le suivant:

PETSPEED est compatible avec BASIC interprété mais BASIC seulement. Pas avec des ROM d extensions comme EDEX, SUPERBASIC ou MASTER. Au contraire, DTL l'est. En fait DTL est plus proche de l'interpréteur que PETSPEED: ceci explique qu'il soit plus lent, mais lui permet en présence d'une instruction d'extension, de faire appel à l interpréteur.

Compatibilité entre versions

Il y a une version de DTL pour chaque machine 3032 4032 8032 et 8096. Un programme compilé sur 8032 passe sur 4032 pas sur 3032.

Il n y a qu une version PETSPEED mixte 4032/8032. Les programmes compilés passent de 4032 à 8032 et inversement. Il n y a pas compatibilité avec le PETSPEED 3032.

Clés électroniques

Chacun des compilateurs est protégé par une clé électronique.

Les programmes compilés par PETSPEED n ont en revanche pas besoin de clé.

Au contraire les programmes compilés par DTL ont besoin soit de la clé de compilation soit d'une clé d'exécution (coût 500F). Ceci est intéressant pour les développeurs de logiciels qui peuvent ainsi vendre des programmes compilés protégés.

Le TINY BASIC COMPILER

Nous ne citons ce compilateur que pour mémoire. Certes il a des qualités: il est moins cher (300F) a une version pour VIC et pour PET 8K (il est écrit en BASIC); c est probablement lui le plus rapide car il produit du véritable code de 6502. mais il n est pas compatible avec le BASIC Commodore normal ce qui à nos yeux. lui enlève beaucoup d'intérêt.

De fait son nom l indique c'est un tiny BASIC.

Et si vous voulez l utiliser il faudra d abord pratiquement compiler votre programme à la main:

. les expressions arithmétiques autorisées n ont qu une opération. donc il faut décomposer.

. il n y a que 26 variables (une seule lettre dans le nom)

. pas de FOR

. pas de tableaux ni matrices

. seul IF autorisé de la forme IF A GOTO ou GOSUB n qui teste si A est nulle ou non.

Nous pensons qu'il suffit de citer ces restrictions.

CONCLUSION

Si l'on exclut le Tiny Basic Compiler on voit que le choix entre les deux compilateurs professionnels n est pas facile. Notons que ce ne sont pas de vrais compilateurs: en fait ils procèdent comme un PASCAL classique avec traduction dans un code intermediaire et interprétation de ce code.

Sur la base de la rapidité. nous donnons la préférence à PETSPEED mais ce jugement peut être inversé pour l'utilisateur qui a besoin de la compatibilité avec EDEX ...

DTL est plus économe en place mémoire et en encombrement des fichiers sur disque.

Nous n avons pas parlé de la vitesse de compilation car elle n est pas un critère important. En effet. en principe on ne compile qu une fois pour exécuter de nombreuses fois. DTL est environ deux fois plus rapide que PETSPEED (attention! à l exécution c est plus que le contraire) mais les temps de compilation de nos essais ont tous été tres brefs.

Le comportement différent des clés électroniques peut jouer en fonction des différentes catégories d utilisateurs: DTL permet de fournir des programmes compilés protégés PETSPEED donne au contraire des programmes libres (mais rien n'empêche l'auteur de les protéger autrement).

Revenons sur les gains en vitesse. plus grands avec PETSPEED mais pas si grands qu on l entend dire. La raison en est que nos programmes d'essai étaient assez dépouillés et optimisés d avance. En fait l intérêt du compilateur est de vous permettre d écrire en BASIC commenté structuré sans chercher d astuces tout en n'étant pas pénalisé en vitesse d exécution.

Daniel-Jean DAVID

... Let There Be Light - Pen ...

Toute la lumière sur les crayons lumineux.

Trois choses dans cet article: d'abord, comment vous bricoler un crayon lumineux connectable au port cassette d'un CBM ou d'un VIC. Ensuite le banc d'essai de deux crayons lumineux du commerce: l'ALPHATRONIC, prévu pour un CBM mais adaptable au VIC et le STACK, prévu pour le VIC.

Bricolage d'un crayon lumineux

Ce bricolage n'est a priori pas difficile: il suffit d'un phototransistor, dont le signal convenablement amplifié est envoyé à un port d'entrée de l'ordinateur.

Au point de vue logiciel, dans les afficheurs de grosses machines prévues pour recevoir un crayon lumineux, voilà comment cela marche: il se trouve qu'un point élémentaire de l'écran n'est allumé que pendant que le spot qui balaie l'écran est en ce point (c'est la persistance des impressions lumineuses qui vous fait croire que l'image est toujours éclairée). Donc le phototransistor n'est éclairé qu'au moment où le balayage passe dessus. Selon l'endroit où vous placez le crayon lumineux, il s'écoulera un temps différent entre le début du balayage et le moment où le spot atteint le crayon lumineux (ce temps est nul si le crayon lumineux est au coin supérieur gauche de l'écran et il est maximum si le crayon lumineux est en bas à droite). On conçoit que la mesure de ce temps permet au système d'en déduire les coordonnées.

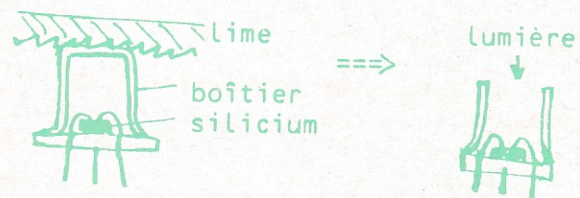
C'est précisément ainsi que fonctionne l'entrée crayon lumineux du VIC et le boitier VIC6561 a deux registres pour contenir les coordonnées du crayon lumineux. Mais ce fonctionnement suppose une très grande sensibilité du phototransistor, une très grande finesse de sa zone sensible et une très grande rapidité de réponse car le temps de balayage est rapide.

Une difficulté vient de ce que la sensibilité ne doit pas être trop grande à cause de la lumière parasite. Il s'ajoute les difficultés dues à la distance entre le verre de l'écran et la couche phosphorescente. C'est pourquoi nous n'espérons pas qu'un crayon lumineux bricolé puisse fonctionner en coordonnées élémentaires et nos essais de connexion au VIC l'ont montré.

En revanche on peut tout à fait espérer que le crayon lumineux sera sensible à un pavé (une maille remplie d'un espace en "reverse"). Ceci est très convenable pour une application de type menu. On affichera un pavé en face de chacun des choix proposés et pour savoir si le choix no. N est sélectionné, on fera clignoter le pavé correspondant. Si le signal du crayon lumineux "suit" les clignotements, c'est que c'est bien le choix N qui a été fait.

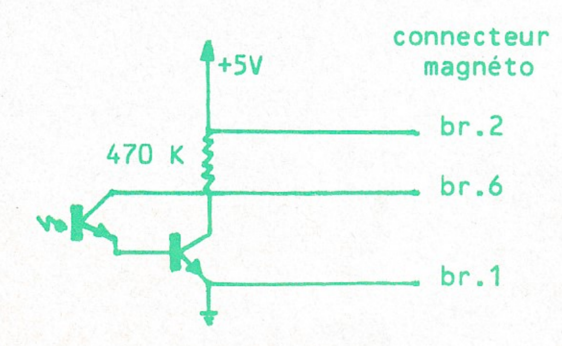
Construction

Nous avons employé comme phototransitor un transistor BC109 en boîtier métallique, dont nous avons supprimé le couvercle à la lime:



Le "photo"transistor est monté en Darlington avec un 2ème BC109. La sortie est envoyée sur l'entrée du connecteur magnéto. On pourrait aussi bien l'envoyer sur une entrée du port utilisateur. Mais comme de toutes façons il faut, sur CBM, récupérer le +5V sur le connecteur

magnéto, notre choix permet de n'avoir qu'un connecteur en jeu.



On forme autour du "photo"-transistor un petit tube en scotch noir qui permet d'isoler la lumière. Si l'on arrivait à faire un tube assez fin on pourrait être sensible aux pavés d'un quart de maille de GR 4000. Il y a un compromis à trouver entre le réglage d'intensité de l'affichage, la longueur du tube et l'intensité de la lumière ambiante (il peut être nécessaire d'éteindre). Il faut aussi que la glace de l'écran soit propre.

Pour effectuer ces réglages vous entrez le programme suivant qui affiche un pavé et imprime la valeur lue au crayon:

```
10   PRINT"Clr↓ ↓ ↓ ↓ ↓ → → → →
→ → Rvs Sp"
20   K = PEEK (A) AND B
30   PRINT "Home"K : GOTO 20
```

avec:

	CBM	VIC
A	59408	37151
B	16	64

Si l'impression ne varie pas quand vous amenez le stylo sur le pavé c'est que rien n'est décelé: essayez d'augmenter la luminosité d'écran.

Essayez de voir quelle est la précision de positionnement qu'il faut et jouez sur les réglages pour essayer de l'augmenter.

Conclusion

Le bricolage que nous donnons est à la fois satisfaisant et limité: il permet, à peu de frais, de se constituer un crayon lumineux capable de repérer un pavé sur l'écran. Mais il ne peut que cela et est donc utilisable dans des applications de type menu. En revanche il n'est pas assez précis et le système de clignotement ne permettrait pas un logiciel assez rapide pour, par exemple, définir un trajectoire sur l'écran. Un élément défavorable est aussi la distance trop grande entre la glace de l'écran et la surface phosphorescente.

Banc d'essai du crayon lumineux ALPHATRONIC

Le crayon lumineux ALPHATRONIC vaut environ 500F. L'exemplaire essayé nous a été prêté par la société SIDEG, que nous remercions.

Le crayon lumineux est bien présenté: il est relié par un fil à un boîtier qui se relie au port utilisateur. Le boîtier est surmonté d'un potentiomètre manoeuvrable par tournevis pour le réglage de sensibilité.

Il n'y a aucune notice: ce qui en tient lieu, c'est le listing d'un programme de démonstration fourni sur cassette. Ne pas oublier de réclamer cette cassette: sans elle le crayon lumineux n'est pas utilisable.

Pour notre exemplaire, nous avons eu quelques difficultés de chargement. Nous avons pu toutefois reconstituer le listing. Les bits du port utilisateur concernés sont 0, 1 et 2. Le point est qu'il faut déclencher le stylo en mettant à 1 les bits 1 et 2 (qui sont en sortie). Le bit 0 (en entrée) indique si le crayon "voit" un pavé éclairé ou non: il est à 1 si éclairé, à 0 sinon.

La cassette fournit un petit programme de réglage de la luminosité et du potentiomètre qui est analogue à celui que nous avons donné ci-dessus:

```
 5   PO=59457:(CBM)/37136 (VIC):
PD=PO+2
10   POKE PD,6
20   POKE PO,4 : POKE PO,6
30   PRINT PEEK(PO) AND 1
40   GOTO 20
```

La cassette fournit d'autre part un exemple de programme pour déceler un pavé allumé sur l'écran dans une application de type menu. La méthode suivie est celle du clignotement, sous une forme assez élaborée puisqu'il faut que le stylo ait suivi au moins 4 clignotements pour qu'on considère un pavé comme certain.

Le problème est que, du moins en ce qui concerne l'exemple que nous avons essayé, le stylo Alphatronic nous a semblé d'une sensibilité un peu faible: il a fallu pousser à fond le potentiomètre et augmenter fortement la luminosité de notre écran pour avoir quelque chose. A ce moment là, le système n'est capable de déceler que des pavés pleine maille. Nous avons constaté sur notre exemplaire qu'il n'était pas capable de déceler les pavés quart de maille de GR4000.

Conclusion

Le crayon lumineux ALPHATRONIC est un achat valable, pas trop cher (500F tout de même) pour l'utilisateur qui veut un stylo lumineux tout fait. Mais nous avons vu qu'il souffre des mêmes limitations que notre stylo bricolé.

Il est adaptable au VIC, mais pour le VIC, le produit de l'essai suivant est préférable. L'Alphatronic ne se justifie donc que si l'on a besoin de la compatibilité VIC/CBM.

Banc d'essai du crayon lumineux STACK

Le crayon lumineux STACK est spécialement adapté au VIC. Il se branche sur le connecteur de jeux et il utilise complètement la possibilité du 6561 de noter à tout moment les coordonnées à repérer. Avec le programme suivant, il suffit de promener le crayon sur l'écran pour repérer des coordonnées:

```
10   X=PEEK(36870) : Y=PEEK(36871)
20   PRINT X,Y : GOTO 10
```

Il faut en fait se placer dans une zone claire de l'écran, donc les mesures seront difficiles dans la zone du bord sauf si la couleur choisie est claire. Sur le fond, s'il est clair, il n'y a aucune difficulté.

Les coordonnées mesurées vont, si l'on exclut le bord, de 45 à 139 en X et de 36 à 124 en Y. On peut à peu près déceler les coordonnées d'un pavé d'1/8ème de maille (1/2 en X car le résultat est toujours impair) et 1/4 en Y.

Le stylo est muni le long du corps d'un contact métallique tel que si l'on touche ce contact avec le doigt, le fait peut être décelé par le VIC (par IF PEEK(37137) AND 16) et donc on peut décider que le programe ne relèvera des coordonnées que s'il y a contact. Cela permet à l'utilisateur de signaler quand il pointe sur le point voulu.

Conclusion

Le crayon lumineux STACK est un excellent achat (en Angleterre 25 livres soit, rendu en France, environ 350 francs). Il exploite à fond les possibilités du 6561 du VIC et pourrait être à l'origine d'applications graphiques spectaculaires ou de jeux passionnants. Notons qu'il est fourni avec une notice adéquate (in english of course!).

Conclusion générale

Des trois possibilités essayées aujourd'hui, le stylo STACK pour le VIC nous a paru le plus séduisant. C'est dû en partie au fait que le hardware du VIC a ce qu'il faut pour exploiter un stylo lumineux. Nous avons été assez déçus du fait que le stylo Alphatronic ne fasse guère mieux que notre bricolage. Ce dernier a le mérite du bas prix de revient; il peut être rendu esthétique en enfermant le tout dans le corps d'un vieux stylo à bille (assez gros).

Daniel-Jean DAVID

VIC à BRAC

Erratum

Comme nous le signale Marc FROIDEVAL, que nous remercions, le programme du no.6 p.32 est faux en ce qui concerne les déplacements à droite (JOY3). D'abord c'est avec 128 (bit 7) et non 32 que doivent se faire les comparaisons. Ensuite, le bit 7 du port B doit être mis en entrée par POKE 37154,127 car il est en sortie en mode normal. Bravo Commodore d'avoir mis JOY3 à part!

Lorsque vous avez fini d'utiliser le JOYSTICK, il faut remettre ce bit en sortie par POKE 37154,255 sinon le clavier se lira incomplètement.

La même erreur se retrouve dans "La pratique du VIC vol.1" pp.128 et 162. En revanche, le programme de La Commode no.6 p.46 est correct.

Voici une version corrigée du programme de la p.32:

```
10 PA=37151:PB=37152
15 POKEPB+2,127
20 K=11:L=11:GOSUB1000
30 IF (PEEK(PA) AND 4)=0 THEN L=L-1
40 IF (PEEK(PA) AND 8)=0 THEN L=L+1
50 IF (PEEK(PA) AND 16)=0 THEN K=K-1
60 IF (PEEK(PB) AND 128)=0 THEN K=K+1
70 IF (PEEK(PA) AND 32)=0 THEN PRINT
CHR$(230)CHR$(157);
80 GOSUB 1000:GOTO 30
1000 IF L<1 THEN L=1
1010 IF L>23 THEN L=23
1020 IF K<1 THEN K=1
1030 IF K>22 THEN K=22
1040 POKE 781,L-1:POKE 782,K-1:POKE 783,0
1050 SYS 65520: RETURN
```

Le générateur de caractères et les extensions

Sur le VIC de base, pour passer en minuscules, il suffit de faire POKE 36869,242. Essayez sur un 16K... bizarre non? Mais tout n'est pas perdu, voici les chiffres corrects:

POKE	base	16K	effet
36869	240	192	majuscules
	241	193	id. inversé
	242	194	minuscules
	243	195	id. inversé
52/56	20	86	réserve mém. 192 carac.
	24	90	id. pour 128 carac.
	28	94	id. 64 car.
	30	96	val. normale

SEUL PROBLEME... c'est que les POKE 36869,253 à 255 n'ont pas d'équivalents pour le 16K ;ou alors on peut les faire... en utilisant un VIC de base. On se demande alors pourquoi mettre une cartouche!

Pour avoir à tous les coups les minuscules à l'écran il suffit de faire:

```
10 P=242 : IF PEEK(36869)=192 THEN
P=194
20 POKE 36869,P
```

Francis ESTEVE

Caractères accentués à l'écran

Créer des caractères est relativement facile, quoique pas trop évident pour un débutant. Créer des caractères accentués, si chers à notre langue, c'est bien... encore

faut-il savoir les mettre au bon endroit, c'est à dire avec les minuscules... alors que le VIC au départ est en majuscules. Voici un programme très simple (on n'y définit que le é, mais les autres accents sont sur le même modèle):

```
6000 PRINT"COMBIEN DE CARACTERES VOULEZ-VOUS:"
6010 PRINT"1=(64) 2=(128) 3=(192) :"
6030 GETZ$:IFZ$=""GOTO6030
6040 ON VAL(Z$)GOTO6050,6070,6090
6050 MM=255:PP=28:GOTO7000
6070 MM=254:PP=24:GOTO7000
6090 MM=253:PP=20
7000 POKE646,10:REM MODE MULTICOLORE
7010 PRINT"▯   L‾● ♠——‾/|●":REM = LES ACCENTS
7020 POKE646,2:REM MODE COULEUR
7030 POKE36869,MM:REM GEN. CARACT. EN MEV/RAM
7040 POKE52,PP:POKE56,PP:CLR:REM RESERVE MEMOIRE
7050 DD=PEEK(52)*256:REM ADRESSE DEPART MEV. RESERVEE
7060 GG=34816-DD:REM MINUSCULES
7070 FF=7679:REM FIN MEV/RAM
7080 FORI=DDTOFF:POKEI,PEEK(I+GG):NEXT:REM CHARGE GEN. CAR. EN MEV/RAM
7090 FORC=DDTODD+7:READA:POKEC,A:NEXT:REM NOUVEAU CAR.
7100 DATA8,16,60,66,254,64,60,0:REM E AIGU
7110 PRINT"▮ R@ALISATION"
```

Tout ce qui est important à savoir est écrit en REM. Il faut simplement rajouter que nous avons mis le titre en mode multicolore pour montrer qu'on peut le combiner avec la génération de caractères.

Ah oui, j'allais l'oublier... le é se fait en appuyant sur le .

Qui nous trouvera comment générer des caractères spéciaux pour un 16K? Cela m'intéresse au plus haut point. Imaginez un traitement de texte sans caractères accentués à l'écran, ce n'est pas toujours très commode pour les nouveaux!

Francis ESTEVE

Nouveau message d'erreur

J'ai eu droit à un:

```
FILE DATA
ERROR IN ...
```

lors d'un INPUT 1,BB,MG,MD, le premier élément étant une chaîne de caractères. C'est évidemment le **fameux** :

```
BAD DATA
ERROR IN ...
```

On voit que le VIC est intelligent!

Francis ESTEVE

Sauvegarde depuis l'adresse voulue

Voici une méthode pour sauver des octets langage machine contenus entre A et B. Si A = LA + 256 * HB:

```
POKE 186,1 : POKE 780,0
POKE 781,LB+1 : POKE 782,HB
POKE 0,LA : POKE 1,HA : SYS 63093
```

que l'on chargera par LOAD "",1,1. Le programme résident ne sera pas écrasé à condition de restaurer les pointeurs (45,46) et (47,48).

Hervé LE MARCHAND

Déplacement du générateur de caractères

Selon "VIC REVEALED" p.121, on peut déplacer le générateur de caractères du VIC à son gré par bloc de 2K entre $1000 et $3000. Cela serait utile avec une extension 8K ou 16K. Suit un exemple pour le faire débuter à $2000: après de nombreux essais, je dois conclure que cette p.121 contient juste un lot d'erreurs typographiques quelque peu supérieur à la moyenne du livre. La seule solution est de déplacer le BASIC. Cela implique un chargement en deux parties, ce qui est souvent exaspérant. En utilisant le tampon clavier, le chargement et l'exécution s'enchaînent

automatiquement avec "SHIFT/RUN".

Si on veut faire débuter le basic après la carte des caractères programmables, par exemple en DE = L + 256 * H, on fera:

```
10 POKE 44,H : POKE 46,H
:POKE 48,H : POKE 50,H
20 POKE 43,L+1 : POKE 45,L+3
:POKE 47,L+3 : POKE 49,L+3
30 A = H * 256 : POKE A,0
:POKE A+1,0 : POKE A+2,0
40 POKE 198,7 : POKE 631,76
:POKE 632,207 : POKE 633,13
50 POKE 634,82 : POKE 635,213
:POKE 636,13 : NEW
```

Les lignes 40 et 50 mettent LOAD:RUN dans le tampon clavier.
Le programme suivant est alors chargé puis exécuté.

Hervé LE MARCHAND

Copie du générateur de caractères

Dans les jeux assez sophistiqués, il est nécessaire de créer ses propres caractères. Il faut cependant conserver une partie des caractères normaux pour les commentaires et les décors. On doit souvent au début du programme transférer les 128 premiers caractères, soit 128x8 octets, dans le nouveau générateur, les nouveaux caractères commençant à partir du code écran 128. Cette opération prend un temps non négligeable. On peut la réaliser instantanément en appelant une routine des ROM.

Exemple : soit à déplacer les 128 premiers caractères, soit 4 pages, à partir de l'adresse:
33792 = 132 * 256 vers:
7168 = 28 * 256 (attention, on commence par le haut : ce sont les adresses qui suivent la dernière adresse transférée):

```
POKE 91,132: POKE 90,0: POKE 88,0
POKE 782,0: POKE 781,4+1: SYS 50163
```

Le nouveau générateur débutera en 7168 - 4 * 256 = 6144 en faisant POKE 36869,254.

Hervé LE MARCHAND

Création de "sprites" pour CBM 64

Des "esprits" non pour faire tourner les tables mais pour animer l'écran de votre 64.

Voici pour tous ceux qui excellent dans la programmation des jeux un programme bien pratique, puisqu'il permet de créer des sprites (figures programmables et mobiles sur tout l'écran; voir l'article sur le CBM 64 dans le numéro 6 de La Commode) sans s'occuper des données à fournir sur les lignes de DATA.

Pour cette création, il vous suffit de dessiner sur l'écran le motif à enregistrer et votre CBM 64 se charge de calculer et placer les résultats de ses cogitations sur les lignes de DATA utilisables à tout moment dans un programme.

Marche à suivre

1 - Après le RUN traditionnel, la machine vous présente le menu des possibilités de création et d'édition de votre sprite (lignes 150 à 270).

2 - Puis, l'ordinateur affiche le cadre de 24x21 mailles dans lequel vous pouvez laisser aller votre sens artistique.

Le cadre se présente ainsi:

```
pointeur
        1 2 3  ...  222324
      1                    1
      2                    2
      .                    .
      .                    .
     20                   20
     21                   21
        1 2 3  ...  222324
```

Lorsque vous utilisez une des mailles de la 24ème colonne, le pointeur disparaît de l'écran mais revient aussitôt, dès que la touche "RETURN" ou "CRSR ==" ou "DEL" est employée. Ceci a pour but d'éviter que le pointeur cache la maille qui vient d'être remplie ou vidée.
Le pointeur qui apparaît en haut à gauche du cadre vous permet un repérage instantané de la position à laquelle vous vous trouvez.
Toutes les touches présentées dans le menu sont à répétition. Ceci est obtenu par POKE de 128 à l'adresse 650 (ligne 120), identique à celle du VIC (voir le no.4 de La Commode sur les adresses strtégiques du VIC). Bon nombre d'adresses sont compatibles avec celles du CBM 64, ce qui constitue un avantage certain.

3 - Une fois votre mini-oeuvre achevée, tapez sur F, puis patientez un instant: votre machine favorite transforme votre dessin en une suite de nombres qu'elle place de 832 à 894 (tampon cassette).
Ceci fait, deux questions apparaissent sur l'écran:

a) ligne 1040:
"no de la première ligne ?"
Il s'agit de fournir le premier numéro de ligne BASIC correspondant au début des lignes à créer. Cette valeur doit être supérieure à 1700 (fin du programme).

b) ligne 1060: " Pas ?"
Donnez ici le pas entre deux lignes. Par exemple, si vous désirez aller de 10 en 10, tapez 10.

Puis, les lignes se composent sous vos yeux. La technique utilisée pour enregistrer les nouvelles lignes est basée sur les possibilités du tampon clavier (voir lignes 1070 à 1220 ainsi que le no.5 de La Commode, pages 27-28).

4 - Listez les nouvelles lignes; exemple: LIST 1701-

5 - Effacez tout le programme par NEW (sauvez-le avant si vous venez de le taper !)

6 - Enregistrez les lignes nouvellement créées en remontant sur l'écran à l'aide de la touche "CRSR ^ ", puis tapez autant de "RETURN" qu'il y a de lignes.

Voilà, le tour est joué! Vous avez en mémoire centrale uniquement ces lignes habituellement si pénibles à composer manuellement.

7 - Sauvez ces quelques lignes au cas où une fausse manoeuvre interviendrait par la suite.

8 - Pour les intégrer à un programme, il faut:

a) les lister en veillant à ce qu'elles restent toutes sur l'écran.

b) charger votre programme, remonter sur la première ligne de DATA (encore présentée sur l'écran) et enregister toutes les lignes au moyen de la touche "RETURN".

Voilà, j'espère qu'avec ce programme, les calculs fastidieux qui mènent à ces fabuleux "sprites" ne seront plus un obstacle pour profiter des possibilités graphiques étendues du CBM 64.

Pierre Ménétré

```
100 REM *** CREATION DE  SPRITES ***
110 REM ***      POUR CBM 64      ***
120 DIM D(21,24),CA(9):POKE 650,128
130 FOR N=1 TO 9
140 READ CA(N):NEXT:C$=CHR$(166)
150 PRINT ""SPC(10)"CREATION DES SPRITES
160 PRINT SPC(10)"******** *** *******
170 PRINT "COMMANDES DISPONIBLES :
180 PRINT "1 - TOUCHES CRSR : MOUVEMENTS
190 PRINT SPC(19)"DU CURSEUR
200 PRINT "2 - E              : ECRITURE
210 PRINT "3 - TABULATEUR     : EFFACEMENT
220 PRINT "4 - DEL            : ANNULATION
230 PRINT "5 - RETURN         : RETOUR CHARIOT
240 PRINT SPC(19)"+ LINE FEED
250 PRINT "6 - F               : FIN DU DESSIN
260 PRINT " ' C ' POUR COMMENCER"
270 GET A$:IF A$<>"C" GOTO 270
280 PRINT "";:GOSUB 1570:PRINT:GOSUB 1530
290 FOR N=1 TO 21
300 A$=RIGHT$(STR$(N),2)
310 PRINT A$SPC(24)A$
320 NEXT
330 GOSUB 1530:GOSUB 1570
340 C=1:L=1:PRINT ""C$;
350 GET A$:IF A$="" GOTO 350
360 A=ASC(A$):FOR N=1 TO 9
370 IF A=CA(N) GOTO 390
380 NEXT:GOTO 350
390 IF N<>7 THEN CP=0
400 ON N GOSUB 450,510,570,630,690,690,770,830,910
410 GOTO 350
420 :
430 MOUVEMENT EN BAS
440 :
450 IF L=21 THEN RETURN
460 GOSUB 1620
470 PRINT ""C$;:L=L+1:RETURN
```

```
480 :
490 MOUVEMENT A DROITE
500 :
510 IF C=24 THEN RETURN
520 GOSUB 1620
530 PRINT "▊"C$;:C=C+1:RETURN
540 :
550 MOUVEMENT EN HAUT
560 :
570 IF L=1 THEN RETURN
580 GOSUB 1620
590 PRINT "▐"C$;:L=L-1:RETURN
600 :
610 MOUVEMENT A GAUCHE
620 :
630 IF C=1 THEN RETURN
640 GOSUB 1620
650 PRINT "▊"C$;:C=C-1:RETURN
660 :
670 FORMATION/EFFACEMENT DU DESSIN
680 :
690 GOSUB 1620
700 IF A=69 THEN PRINT "▓ █";:D(L,C)=1
710 IF A=32 THEN PRINT " ";:D(L,C)=0
720 IF C=24 THEN RETURN
730 PRINT C$;:C=C+1:RETURN
740 :
750 ANNULATION
760 :
770 IF CP THEN CP=0:C=C-1:D(L,C)=0:PRINT "▊ ▊▊"C$;:RETURN
780 IF C=24 THEN CP=1:D(L,24)=0:PRINT "▊"C$;:RETURN
790 GOSUB 630:D(L,C)=0:RETURN
800 :
810 RETOUR CHARIOT + LINE FEED
820 :
830 GOSUB 1620:IF C<>24 THEN GOSUB 1630
840 C=1:IF L<21 THEN L=L+1
850 PRINT "▓▓";
860 FOR N=1 TO L:PRINT "▓";:NEXT
870 PRINT "▊▊"C$;:RETURN
880 :
890 ENREGISTREMENT DES DONNEES
900 :
910 PRINT "▓▓▓"SPC(29)"OK, JUST
920 PRINT "▓"SPC(29)"A MINUTE
930 PRINT "▓"SPC(29)"PLEASE ..."
940 P=832:POKE 650,0:FOR N=1 TO 21
950 FOR K=0 TO 2
960 R=0:FOR I=1 TO 8
970 R=R+D(N,K*8+I)*2↑(8-I)
980 NEXT I:POKE P,R:P=P+1
990 NEXT K,N
1000 :
1010 REM MISE SOUS FORME DE DATA
1020 :
1030 N=832:D=N+62
1040 INPUT "▐▓▓NO DE LA PREMIERE LIGNE ";LC
1050 IF LC<1700 GOTO 1040
```

```
1060 INPUT "█PAS ";PA:POKE 831,PA
1070 GOSUB 1210:GOTO 1100
1080 IF POS(0)>35 THEN T=1:GOTO 1130
1090 PRINT ",";
1100 B$=STR$(PEEK(N))
1110 PRINT RIGHT$(B$,LEN(B$)-1);
1120 IF N<>D THEN N=N+1:GOTO 1080
1130 PRINT:PRINT "1180 N="N":D="D":LC="LC
1140 IF T THEN PRINT "GOTO 1180█";:GOTO 1170
1150 PRINT "END█";
1160 REM 3 RETURN DANS LE TAMPON CLAVIER
1170 POKE 198,3:POKE 631,13:POKE 632,13:POKE 633,13:END
1180 *** LIGNE MODIFIEE LORS DE L'EXECUTION
1190 GOSUB 1210:GOTO 1100
1200 *** DEBUT DE LA NOUVELLE LIGNE
1210 PRINT "███"LC"DATA ";:LC=LC+PEEK(831)
1220 RETURN

1500 :
1510 PRESENTATION DES CHIFFRES
1520 :
1530 PRINT "   ";
1540 A$="1234567890"
1550 PRINT A$A$LEFT$(A$,4):RETURN
1560 :
1570 A$="         ":PRINT "    ";
1580 PRINT A$1A$2;:RETURN
1590 :
1600 POSITIONNEMENT DU POINTEUR
1610 :
1620 PRINT "█";
1630 IF D(L,C) THEN PRINT "█";
1640 PRINT " ██";:RETURN
1650 :
1660 CODES ASCII DES COMMANDES DISPONIBLES
1670 DATA 17,29,145,157,69,32,20,13,70
1680 :
1690 LIGNES DATA CREES PAR LE PROGRAMME
1700 :
```

Utilisation des Joystick sur le VIC

Après avoir acheté la Commode numéros 4 et 5 au SICOB, j'ai été attentif aux adresses pour l'utilisation du joystick que vous m'avez indiquées. (VIC 20).

Toutefois, il faut préciser que ces adresses correspondant aux ports A (37137) et aux ports B (37152) dépendent des valeurs des bits des D.D.R. (?) des V.I.As correspondants.

Si le D.D.R. du port A d'adresse 37139 est positionnée correctement à la mise sous tension, par contre ça n'est pas le cas du D.D.R. du port B d'adresse 37154, me semble-t-il.

Il faut donc positionner le bit 7 du DDR du port B, mais attention, il faudra le réinitialiser après utilisation car il sert aussi au clavier (risque de résultats improbables).

Je ne sais si vous avez déjà indiqué cela précédemment ; pour ma part, c'est une découverte et je vous propose un programme que j'utilise en démonstration aux écologistes amis de la nature de l'Uzège (UZES).

Etant un adepte de la programmation structurée, mon programme est décomposé en structure et traitements.

Les valeurs que mon joystick me donne pour PEEK(37152) + PEEK(37137) sont:

353	369	241
357	373	245
349	365	237

FIRE seul: PEEK(37137) < 100

```
 0 REM VOL DE L'OISEAU CHANTANT
10 GOSUB 100
20 GOSUB 200 : IF ENCORE THEN 20
30 GOSUB 500

100 REM INITIALISATION
110 ? " CLR"
190 RETURN

200 REM MOUVEMENT ET CHANT
210 POKE 37154,127
220 X = PEEK(37152)
230 POKE 37154.255
240 Y = PEEK(37137) + X
250 ? " b b b ←←← " ;
260 IF Y<= 237 THEN ? "→↓ " ; :
    GOTO 350
270 IF Y<= 241 THEN ? "→↑ " ; :
    GOTO 350
280 IF Y<= 245 THEN ? "→  " ; :
    GOTO 350
290 IF Y<= 349 THEN ? "↓← " ; :
    GOTO 350
300 IF Y<= 353 THEN ? "↑← " ; :
    GOTO 350
310 IF Y<= 357 THEN ? " ←" ; :
    GOTO 350
320 IF Y<= 365 THEN ? "↓" ; :
    GOTO 350
330 IF Y<= 369 THEN ? "↑" ; :
    GOTO 350
340 REM OISEAU
350 ?"( 0 )←←←" ;
360 FOR I = 1 to 55 : NEXT
370 ?"( 0 )←←←" ;
380 REM CHANT
390 POKE 36878.15
400   FOR M = 254 TO 240 +
INT(RND(1)*10) STEP-1
410 POKE 36876.M
420 NEXT M
430 POKE 36878,0
440 ENCORE = (Y-X > 100)
490 RETURN
500 REM FIN
510 STOP
```

De 0 à 30 c'est la structure du programme

210 et 230 sont le positionnement et la réinitialisation du DDR du port B

'Encore' correspond à l'appui du bouton "FIRE" tout seul.

Max HAGENBURGER

La Page du 1er avril

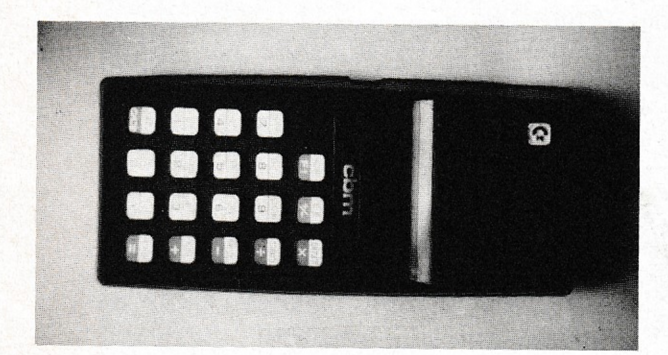
LE FUTUR CBM

Nos informateurs au siège de Commodore Californie ont réussi à nous faire parvenir, par un canal que nous ne pouvons révéler, le document ci-dessous qui représente le prochain modèle de CBM. Comme le montre la photo, il n'a pas encore reçu de numéro définitif.

Ce modèle marque une étape décisive dans la voie de la miniaturisation, compte tenu de la formidable puissance de calcul qui est renfermée dans un si petit volume.

Progrès essentiel pour les utilisateurs qui ne veulent (ou ne peuvent) entrer dans les détails si assommants de la programmation: ce modèle possède un mode de fonctionnement en émulation de calculatrice de poche; dans ce mode, vous obtenez immédiatement le résultat de vos calculs sans avoir à passer par une programmation fastidieuse.

Le Microprocesseur utilisé est un MOS Technology 7529, membre de la série 7500 appelée à succéder à la série 6500.



Les contrePETeries de La Commode

Compatibilité dans la gamme Commodore:

Votre VIC est un PET.

Livraison par bateau d'une carte EXT VIC BUS 4:

Monsieur, votre plaquette
est à quai.

Effets sonores:

Le son du VIC.

Un autre titre pour la rubrique des jeux cousus-mains:

Le coin des jeux.

N.B. Il est d'usage de ne jamais publier le décodage des contrePETeries. Il est en revanche possible que nous publiions un programme d'intelligence artificielle qui effectue un tel décodage.

ANNONCE IMPORTANTE

A compter du numéro 8 -oui, le prochain- La Commode passe à 96 pages soit une augmentation de 50% puisque la périodicité reste trimestrielle. Ceci pour une augmen-

tation de prix bien moindre.

En même temps, La Commode se décloisonne: il n'y aura plus que du Commodore, mais aussi de l'ATARI et de l'ORIC. LA COMMODE RESTE SPECIALISEE (6502 oblige) MAIS ELLE N'EST PLUS EXCLUSIVE ET A DES TIROIRS EN PLUS.

COMMODORE 104

(Pas Peugeot ... Commodore)

Commodore aurait dans ses cartons un modèle qui devrait même sortir à la place du 64 appelé 104, bâti autour d'un microprocesseur 0104: un processeur de la famille 6500 mais à jeu d'instructions microprogrammable, espace adressable de 16 M-octets en 256 bancs de 64K, et horloge 8 ou 16 Mhz.

Toutes les caractéristiques du 104 seront programmables notamment les bancs de mémoire actifs, le format de l'écran et la signification de chaque touche sur le clavier de 112 touches: la mémoire de décodage est en RAM; c'est commode pour l'AZERTY n'est-ce pas?

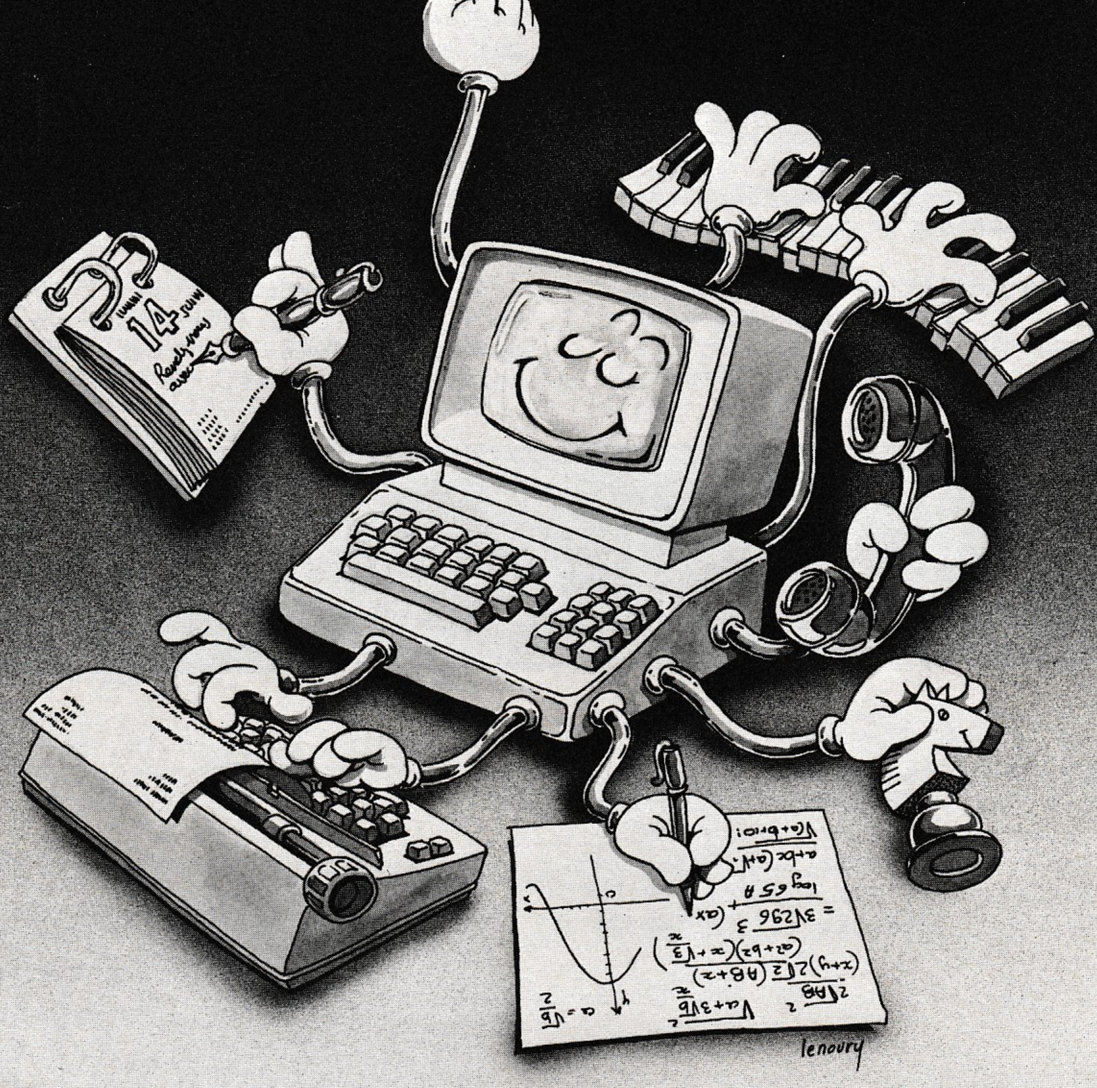
La résolution de l'écran est de 1056 sur 480 points qui peuvent être organisés en caractères de 20x10 à 132x60 en passant par le standard 80x25 qui est ce qu'on a à la mise sous tension. L'écran est géré par un 6502-A consacré à cette tâche.

Il n'y a pas d'interface cassette mais un bus IEEE complet.

Mais la caractéristique essentielle est la micro-programmation qui permet au 0104 d'émuler n'importe quel microprocesseur et donc au système d'émuler n'importe quel microordinateur du marché. Les versions proposées seront PET/CBM en BASIC 2 ou 4.0, 8000 ou 9000, ATARI 800, APPLE II ou III, ZX81 etc...

Ces informations sont bien sûr confidentielles. Elles nous ont été données par le Club des Utilisateurs Britanniques.

MICRO EXPO 83



8e Congrès-Exposition - MICRO-ORDINATEURS

Palais des Congrès - CIP - Porte Maillot - Paris

organisé par :

SYBEX

SYBEX 4, place Félix-Éboué - 75583 PARIS - Tél. : (1) 347.30.20 - Telex : 211 801 F

TOUT SUR LA MICRO INFORMATIQUE

du 14 au 18 juin 1983

L'Exposition :

du 14 au 18 juin 1983

- mardi 14, mercredi 15, vendredi 17, de 9 h 30 à 18 h
- nocturne le jeudi 16, de 9 h 30 à 22 h
- samedi 18 de 9 h 30 à 17 h.

Tous les matériels, toutes les applications des micro-ordinateurs. Que votre motivation soit professionnelle ou personnelle, venez évaluer les matériels, comparer leurs coûts et vous renseigner sur les applications disponibles.

Nouveauté 83 : une animation « logiciel »

(organisée par Sybex et Logiciels & Services).

Une banque de données des logiciels présentés par les exposants à MICRO-EXPO vous guidera, selon vos besoins spécifiques, à travers l'exposition.
Un « grand concours » destiné à récompenser les meilleurs logiciels pour micro-ordinateurs. Les lauréats présenteront leurs logiciels pendant toute la durée de MICRO-EXPO (Renseignements : Logiciels & Services, tél. : (1) 226.11.25).

Et pendant toute la durée de l'exposition, une animation-jeux : venez affrontez l'ordinateur et peut-être... le battre.

Le Congrès

du 13 au 18 juin 1983

Du débutant à l'expert, une occasion unique dans l'année de se former et de s'informer.

Séminaires professionnels
Les micro-ordinateurs : présentation, choix
Les microprocesseurs
Les langages : BASIC, PASCAL, APL
Les nouveaux langages : FORTH, LOGO, PROLOG, C
La télématique
Le traitement de texte
L'entreprise, son système d'information et l'informatique
La comptabilité et ses logiciels
Les bases de données et leurs logiciels
La gestion et le Visicalc
Le système d'exploitation MS-DOS

Conférences Grand Public
Choix d'un micro-ordinateur
Les micro-ordinateurs de poche
Le traitement de texte
L'enseignement assisté par ordinateur - E.A.O.
Les systèmes d'exploitation pour micro-ordinateurs 16 bits
Journées spécialisées, avec démonstrations : — pour les professions juridiques, — pour la médecine, — pour l'agriculture.
Journées des constructeurs : COMMODORE, GOUPIL, HEWLETT-PACKARD, TANDY, THOMSON, VICTOR LAMBDA

INFORMATION

SYBEX

Nom :

Adresse :

Code postal : Ville :

☐ ____ entrées(s) à demi-tarif

☐ Programme détaillé du congrès

4, place Félix-Éboué - 75583 PARIS Cedex 12 - Tél. : (1) 347.30.20 - Telex : 211 801 F

Com. 3/83

Les jeux cousus-main

Les Envahisseurs

```
100 REM************************************
110 REM*          LES ENVAHISSEURS        *
120 REM*    TRADUCTION ET ADAPTATION      *
130 REM*********DEPAYE ERIC  - 1982 -****
140 A=32768:B=40:F=9:F2=9:D8=9:PRINT"▢▓▓▓■■■VOULEZ-VOUS LES INSTRUCTIONS"
150 POKE167,0:GETA$:IFA$=""THEN150
160 IFA$="O"THENPOKE167,1:GOSUB910
170 POKE167,1:PRINT"▢"
180 DATA200,298,520,466,570,580,96,93,525,9:FORI=1TO10:READD:POKED+A,42
    :NEXT
190 RESTORE
200 REM*** -PREPARATION DE L'ECRAN-    ***
210 K=32
220 FORI=A+22*BTOA+25*B-1:POKEI,102:NEXTI
230 FORI=A+22*B+30TOA+23*B-2:POKEI,30:POKEI-B,100:NEXTI
240 FORI=A+23*B+30TOA+24*B-2:POKEI,I-(32768+23*B+29)+48:NEXTI
250 PRINT"▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓■■■■■■■■■■■■■■■■■■■■■■■■■■■■▓BASE▓"
260 REM*** -ALTITUDE DES ENVAHISSEURS-***
270 D=INT(RND(TI)*19+1)*B+A
280 E=E+1:IFE=10THEN660
290 PRINT"▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓▓FLOTTE ENNEMIE :▓"10-E
300 PRINT"▓■▓TOUCHES :▓"F3"▓"
310 REM*** -DESSIN DE L'ENNEMI-         ***
320 POKED,223:POKED+1,98:POKED+2,121:POKED+3,100
330 POKEL,K:L=D
340 FORT=1TO5
350 GETR$:R=VAL(R$)
360 IFR>0THEN380
370 NEXTT
380 IFR=0THEN400
390 GOTO430
400 D=D+1:IFD-7=INT(D/B)*BTHEN420
410 GOTO320
420 POKED,K:POKED+1,K:POKED+2,K:POKED+3,K:GOTO270
430 P=A+22*B+29+R
440 FORT=A+22*B+30TOA+22*B+38
450 IFPEEK(T)<>KTHEN470
460 NEXTT:GOTO780
470 POKEP,K:P=P-B
480 IFPEEK(P)=KTHEN320
490 D8=D8-1
500 POKEP,K
510 P=P-B
520 IFP<ATHENPOKEJ,K:POKEJ+B,K:GOTO320
530 IFPEEK(P)<>KTHEN840
540 POKEP,65:POKEJ,K:J=P
550 D=D+1:IFD-7=INT(D/B)*BTHENPOKEP,K:GOTO590
560 POKED,223:POKED+1,98:POKED+2,121:POKED+3,100:POKEL,K
570 L=D
580 GOTO510
```

```
590 REM*** -TIR DES MISSILES-        ***
600 P=P-B
610 IFP<ATHEN650
620 POKEP,65:POKEJ,K:J=P
630 GOTO600
640 F3=F3+1
650 POKEJ,K:GOTO420
660 PRINT"■LA FLOTTE ENNEMIE EST PASSEE !!!"
670 PRINT"IL RESTE"9-F3"VAISSEAUX ENNEMIS"
680 PRINT:PRINT"POUR ATTAQUER LA BASE..."
690 FORI=1TO10:GETR$:NEXTI
700 IFF3>F4THENPRINT:PRINT"C'EST LE MEILLEUR SCORE ENREGISTRE":F4=F3:GOTO740
710 PRINT:PRINT"LE MEILLEUR SCORE EST ACTUELLEMENT"F4"   TIRS AU BUT":PRINT
720 IFF3=9THENPRINT:PRINT"LA BASE EST SAUVEE;VOUS ETES UN HEROS!!!!!!!"
730 IFF3=8THENPRINT:PRINT"LE DERNIER VAISSEAU A ETE DETRUIT PEDANT SON ATTAQUE"
740 IFF2<2THENPRINT"LES ENVAHISSEURS ONT FAIT DU BON TRAVAIL":PRINT"■TOUT EST P
ERDU!!"
750 PRINT:PRINT"POUR REJOUER,APPUYEZ SUR 'SPACE'..."
760 WAIT59410,4,4
770 CLR:GOTO110
780 PRINT"■VOUS AVEZ UTILISE TOUTES VOS MUNITIONS;":PRINT
790 PRINT9-F3"VAISSEAUX ONT SURVECUS A L'ATTAQUE"
800 PRINT:PRINT"VOUS AVEZ DETRUIT"F3"VAISSEAUX..."
810 GOTO690
820 PRINT"■■■■■■■VOS RADARS VOUS INDIQUENT QUE"
830 WAIT59410,16,16:RETURN
840 H=42:GOSUB880:H=160:GOSUB880:H=102:GOSUB880:H=81:GOSUB880:F=F+1
850 F3=F3+1
860 H=87:GOSUB880:H=46:GOSUB880:H=K:GOSUB880
870 GOTO270
880 POKEP,H:POKE+1,H:POKEP-1,H:POKEP+40,H:POKEP-40,H
890 POKEP+2,H:POKEP-2,H:POKEP+80,H:POKEP-80,H
900 POKEP+39,H:POKEP+41,H:POKEP-41,H:POKEP-39,H:RETURN
910 PRINT"■"
920 PRINTTAB(13)"■DEFENSE DE L'ESPACE■ "
930 PRINT:PRINT:PRINT"■■VOUS ETES L'OFFICIER DES ARMEES":PRINT
940 PRINT"■■DE LA BASE DU ■CENTAURE■...."
950 PRINT:PRINT:PRINT"■■■■DANS QUELQUES INSTANTS,VOUS SUBIREZ"
960 PRINT"■■UNE ATTAQUE D'ENVAHISSEURS VENUS DE L'AUTRE COTE DE L'UNIVERS....
970 PRINT"■■■■■■■■APPUYEZ SUR 'RVS' POUR ■CONTINUER■..."
980 WAIT59410,1,1
990 PRINT"■■■■■■■VOS RADARS VOUS INDIQUENT QUE"
1000 PRINT"■■■9■ VAISSEAUX ENNEMIS APPROCHENT DE"
1010 PRINT"■■LA BASE !!!"
1020 PRINT"■■■VOUS N'AVEZ JUSTE LE TEMPS D'ENGAGER"
1030 PRINT"■■LA BATAILLE AVEC VOS MISSLES               BALISTIQUES;"
1040 PRINT"■■CEPENDANT L'ORDINATEUR N'EST PAS EN     ETAT DE FAIRE FONCTIONNER"
1050 PRINT"■■LES COMMANDES AUTOMATIQUES DE MISE A    FEU DE CES MISSILES..."
1060 PRINT"■■VOUS ETES DONC OBLIGE DE LES COMMANDER  EN MODE MANUEL;"
1070 PRINT"■■POUR LE FAIRE,IL SUFFIT,POUR METTRE A ■■FEU"
1080 PRINT"■■LE MISSILE 1,D'APPUYER SUR '1',ET       AINSI DE SUITE"
1090 PRINT"■■POUR LES AUTRES MISSILES..."
1100 PRINT"■■■■■■■■■■■■BONNE CHANCE !!!"
1110 PRINT"■■■APPUYER SUR '<' POUR COMMENCER."
1120 WAIT59410,8,8
1130 RETURN
```

Les Astuces de Peter JOLLYBUG

Peter Jollybug est de retour pour nous faire profiter de son expérience.

Nostalgie

J'étais en train de déjeuner en famille quand nous nous sommes posé la question : depuis combien de temps je tapais sur mon ordinateur ? Et bien ! celà fait CINQ ans déjà que le PET est chez moi...
Cinq ans de bugs et de nuits pas toujours reposantes. On a changé de sujet de conversation quand le grand-pere nous a expliqué que de son temps "Tout çà. çà marchait au gaz...".

Je ne crois pas que je pourrais m'empçher de continuer à appeler mon PET un PET CBM, pour le gars qui programme chez lui, ça sonne moins bien que PET ; outre les mauvais jeux de mots du début, le nom de PET évoque bien l'animal domestique que l'on couve jalousement.

Et on en a fait, des choses ensemble ! Au grand dam de ces dames, l'apparition du PET en a gaché, des ménages et en a renforcé dans leur action, des célibataires endurcis ! Que de conflits pour les acharnés "qui ne pensent qu'à leurs programmes". Que de moments en société ou tout ce qui se passe autour de soi s'estompe pour laisser la place à la question "pourquoi ce p... de programme se plante à tel endroit?". Tant pis pour ceux et celles qui ne rentrent pas dans la Confrérie des Cinglés de la Programmation (CCP), les seuls couples heureux sont ceux qui ont DEUX ordinateurs ; pourquoi pas un PET et un VIC ?

Le tout premier modèle n'avait que 7 K octets disponibles, c'était le "vieux" 2001. L'extension anglaise appelée PETITE m'a permis de passer un dimanche exaltant à intégrer les K octets supplémentaires DANS LE PET ! Décortiquer la Petite n'était rien, mais rentrer les trois circuits imprimés dans le PET avec deux transformateurs supplémentaires, 29 trous de fixation à percer et quelques 126 soudures à faire pour pousser le 2001 à afficher ses 31 K octets était une autre affaire...

Et puis, l'aspect du PET est devenu plus aristocratique, avec son écran vert et l'encadrement noir. Et puis, il est devenu plus performant avec le CBM, son beau clavier et ses extensions : imprimante et disque. Et puis on s'est désintéressé du particulier avec la série 8000 qui s'adresse aux professionnels. Et tout le petit monde de l'informatique nocturne regrette le vieux 2001 avec son clavier qui requiert l'habileté de la dentelière et son magnétophone intégré si pratique.

Qu'est-ce qu'ils sont devenus, les pionners inconditionnels du PET? Ont-ils tous "CEBEHEMISE" leur PET, opération "coup de fouet" (à l'instar de la 4 CV) qui consistait en son temps à remplacer les ROMs pour bénéficier des améliorations du CBM ?

Je voulais manifester toute ma sympathie à tous ces confrères qui vibrent au même diapason en mettant sous tension leur système chaque soir.

Astuces

Au risque de paraître radoter, je ne pourrai pas terminer mon "papier" sans rappeler quelques astuces dont les programmeurs de PET/CBM raffolent. Que ceux qui les connaissent déjà veuillent bien avoir l'indulgence de considérer ces lignes comme consacrées aux débutants et, s'ils en connaissent de meilleures, qu'ils en profitent pour prendre leur plume et nous en faire bénéficier.

Toutes les adresses fournies ici son celles du CBM 30xx, ce sont également celles du 2001 cébéhémisé. Les correspondances avec l'ancien 2001 sont données en fin d'article.

1. TOUCHE SHIFT : c'est l'astuce la plus connue. L'attente de l'enfoncement de la touche shift se fait par WAIT152,1 et l'attente de son relâchement se fait par WAIT152,1,1 Utile pour des points d'arrêt aussi bien en développement qu'en exécution.

2. INHIBITION DU STOP : il est intéressant d'interdire le STOP au cours de l'exécution d'un programme. Cela se fait par un POKE 144,49 ; on l'autorise de nouveau par un POKE 144,46. Attention, entre ces deux ordres il n'est pas possible de bloquer le déroulement du programme ! A n'utiliser que quand le programme est parfaitement au point... et après l'avoir sauvé sur cassette ou sur disque.

3. BUFFER CLAVIER : dans un GET ou un INPUT, on prend en compte toutes les touches qui ont été enfoncées DEPUIS LA DERNIERE RENTREE AU CLAVIER, même si l'on a tapé sur le clavier inopinément. On supprime cet inconvénient en faisant un POKE 158,0 avant tout INPUT ou GET, systématiquement; ce POKE a pour effet de remettre à zéro le pointeur du buffer du clavier.

4. RESERVATION DE MEMOIRE : pour stocker un programme en langage machine par exemple sans qu'il soit altéré par le BASIC. Il existe déjà une zone disponible entre $033A et $03EF (826 à 1007), zone qui correspond au tampon réservé à l'usage de la deuxième cassette peu employée. Si cette zone est considérée comme insuffisante, on peut réserver autant d'espace que l'on veut en haut de mémoire utilisateur AVANT TOUTE EXECUTION BASIC par l'ordre suivant :

```
POKE 52,XX : POKE 53,YY : CLR
```

Où XX et YY décrivent l'adresse à partir de laquelle la réservation est faite. En terme "machine", XX est la traduction en décimal du poids faible de l'adresse et YY la traduction en décimal du poids fort. Si l'on veut réserver 4 K octets dans un 3032 entre l'adresse $7000 et $8000, il faut traduire $7000 comme suit :

```
poids faible : $00 = 0.
poids fort : $70 = 7*16 = 112
d'où POKE 52,0 : POKE 53,112 :
CLR
```

Le CLR est indispensable pour remettre à zéro certains vecteurs internes liés au BASIC.

Voici un deuxième exemple où l'on veut réserver 1 Koctet dans le haut de mémoire d'un 16 K, donc la zone allant de $4C00 à $5000 :

```
poids faible : $00 = 0
poids fort : $4C = 4*16+12 =76
d'où POKE 52,0 : POKE 53,76 :
CLR
```

5. CURSEUR CLIGNOTANT : il est intéressant de rendre le curseur visible dans l'écran, notamment dans un GET, par exemple. Le "curseur" matérialise par son clignotement la position dans l'écran ou le prochain caractère va s'afficher. On le rend visible par POKE 167,0 et on le supprime par POKE 167,255. Attention ! il peut rester définitivement allumé quand on a cessé le clignotement. On évite cet inconvénient par l'écriture d'un espace avec ou non un ";" selon que l'on veut faire un retour à la ligne ou non.

6. AFFICHAGE RAPIDE : l'affichage des 2000 et 3000 se fait en

attendant le retour du spot dans l'écran. On peut supprimer cette attente par un POKE 59458,62. L'affichage est alors ultra-rapide! Le revers de la médaille est que des parasites se manifestent dans l'écran qui deviennent pénibles si les affichages sont fréquents.

7. NOMBRES ET MID$: se rencontrent heureusement dans une petite astuce bien pratique quand on n'a pas le PRINTUSING sous la main. Il faut savoir que quand on applique la fonction STR$ sur un nombre, un espace est accollé à gauche de la chaîne qui en ressort ; cet espace correspond au signe qui est éludé quand il est positif. Cet espace peut géner certains affichages. D'autre part la fonction MID$ a un comportement particulier si l'on omet le troisième argument; si l'on n'indique pas la longueur de la sous-chaîne tout ce qui est à droite du caractère pointé dans la chaîne est pris en compte. En mélangeant ces deux propriétés on arrive à l'astuce qui supprime élégamment l'espace dans la conversion nombre-chaîne :

```
N$ = MID$ (STR$(N),2)
```

Correspondances

Terminons par les correspondances entre le "vieux" 2001 et le 3000.

DESCRIPTION	2000	3000
TOUCHE SHIFT	516	152
HAUT MEMOIRE	134	52
	135	53
POINTEUR BUF.CLAVIER	525	158
ARGUMENT INHIB.STOP	..	46
	..	49
CURSEUR CLIGNOTANT	...	167

En vous souhaitant toujours beaucoup de plaisir sur votre système, Peter vous transmet toute son amitié qui fait encore partie des choses qu'il n'a pas besoin de programmer pour la manifester.

PETER JOLLYBUG

L'Interface RS 232 du VIC

La présence de cette interface sur le VIC est très intéressante. Il s'agit en effet d'une interface standard immensément repandue dans le monde des ordinateurs. De nombreux périphériques sur le marché en sont munis: la présence de cette interface et surtout la présence dans le système du logiciel qui la gère peuvent vous permettre de connecter de tels périphériques à votre VIC, notamment des imprimantes. Mais surtout l'interface RS232 est l'interface standard des télécommunications entre ordinateurs. Elle vous permettra de relier votre VIC à des réseaux sur lesquels vous pourrez interroger des bases de données.

Quelques caractéristiques de la norme RS 232

Nous ne donnons pas trop de détails sur les communications série, renvoyant pour cela à Daniel Jean DAVID, Les Systèmes à Microprocesseur, EDITEST. La caractéristique fondamentale des communications RS232 est qu'elles sont en série, c'est-à-dire qu'un octet n'est pas transmis d'un coup mais chaque bit est envoyé successivement.

Cela procure une économie de nombre de fils : au lieu de 8 fils pour transmettre tous les bits en même temps, on n'a plus besoin que d'un fil sur lequel les différents bits circulent l'un après l'autre. De fait, il y a un mode simplifié de fonctionnement du RS 232 où il n'y a que trois fils, la masse, donnée émise et donnée reçue.

La norme RS232 fixe deux choses :

- la forme du connecteur standard RS232 qui est une prise "Cannon" 25 broches et la liste des signaux ainsi que le brochage.

- les propriétés électriques : lorsque le signal RS232 est à 1, la tension doit être comprise entre -3V et -12V ; lorsqu'il est à 0, la tension doit être comprise entre +3V et +12V.

Le VIC sort de la norme RS232 sous plusieurs aspects :

1 - certains signaux ne sont pas gérés par le logiciel système comme "ring indicator" - En revanche, les autres signaux sont gérés : il y a deux modes qu'on choisit par programme, le mode simplifié dit "trois fils" où on gère masse, Sin et Sout, et le mode dit "X fils" où on gère aussi les signaux de commande d'un MODEM (organe de connexion à une ligne téléphonique).

2 - Les signaux sont au niveau TTL (entre 0 et +5V) et non entre -12V et +12V et le connecteur est celui du port utilisateur, donc il n'a pas la forme RS232 standard.

C'est pourquoi Commodore commercialise une cartouche appelée VIC 1011 qui se branche sur le connecteur A (connecteur du port parallèle donc, quand vous utilisez le RS232, vous ne pouvez pas disposer du port parallèle) et possède de l'autre côté une prise de forme RS232 standard, ayant les signaux aux niveaux + 12V caractéristiques du RS232.

Voici les brochages :
connecteur RS232 normalisé :
(broches 9 à 17, 19 et 21 à 25 inutilisées).

1 2 3 4 5 6 7 8 9 10 11 12 13
14 15 16 17 18 19 20 21 22 23 24 25

Connecteur A du VIC

broche VIC(A)	Rôle RS232		numéro de broche sur connecteur normalisé	rôle si utilisé en port parallèle
A	masse		1-7	masse
B	donnée reçue	E*	3	CB1
C	donnée reçue	E*	3	PB0
D	RTS demande d'émission	S	4	PB1
E	DTR terminal prêt	S	20	PB2
F	RI bouclage	+	18	PB3
H	DCD détection porteuse	E	8	PB4
J	non utilisé			PB5
K	CTS prêt à émettre	E	5	PB6
L	DSR modem prêt	E	6	PB7
M	donnée émise	S*	2	CB2
N	masse		1-7	masse

notes: E=entrée S=sortie *=les seules lignes de l'interface trois fils (RTS et DTR sont maintenues à 1 dans ce mode)
+=non géré par le logiciel du VIC. Les broches B et C doivent être connectées toutes les deux au signal "donnée reçue"

Différents paramètres programmables

Le problème principal est celui de la synchronisation. Elle est réalisée de deux manières :
1 - on fixe la fréquence des bits
2 - on définit le départ de chaque caractère.

Pour cela, lorsqu'une ligne est au repos, elle est au niveau 1. Le départ est marqué par l'émission d'un bit à 0, dit bit de départ. Ensuite on a les 5 à 8 bits. (nombre programmable) qui forment le caractère. Il y a ensuite éventuellement 1 bit de parité, puis 1 ou 2 bits à 1, dits bits d'arrêt. La ligne reste alors à 1 durant un temps variable jusqu'au prochain caractère. Notez que les bits transmis le sont en commençant par le moins significatif.

Soit à transmettre un A en ASCII (code en 7 bits hexadécimal 41, décimal 65, binaire 1000001). Le bit de parité sera à 1 (parité impaire) d'où le signal transmis en fonction du temps :

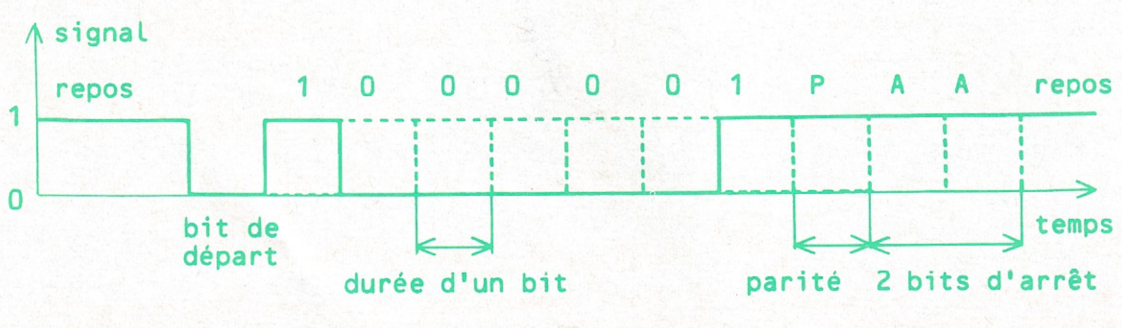
Sur un télétype (terminal très répandu), on fonctionne à 10 car/seconde. Chaque caractère fait 11 bits (départ + 7 + P + deux bits d'arrêt). Il faut donc transmettre à 110 bits/seconde, on dit 110 bauds. Une vitesse plus répandue est maintenant 300 bauds. On sait aller jusqu'à 19200 bauds et même plus (pas le VIC dont l'interface RS232 plafonne à 2400 bauds).

Programmation de l'interface RS 232 du VIC

Les différents éléments programmables sont déterminés dans l'interface RS232 du VIC lors de l'instruction OPEN de la forme :

```
OPEN NFL,2,0,CHR$(V)+CHR$(W)
```

On voit que le numéro de périphérique est 2 et que l'adresse secondaire est 0 ou n'importe quoi. Le numéro de fichier logique est arbitraire. Tout se passe comme si le nom de fichier était formé des



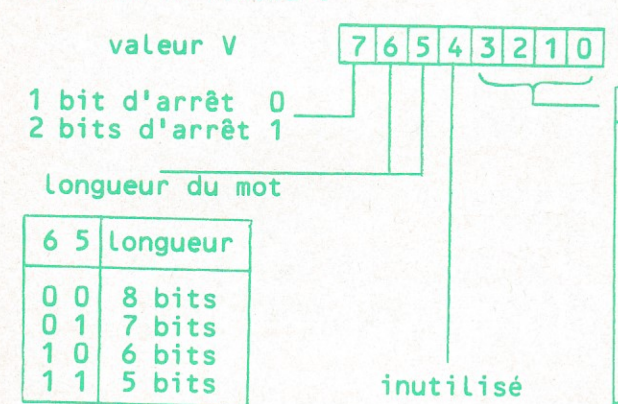
deux caractères de codes V et W (V et W sont à considérer comme des octets, compris entre 0 et 255).

V détermine :

. le nombre de bits d'arrêt : 1 ou 2 (bit 7)
. le nombre de bits de donnée : 5 à 8 (bits 6 et 5)
. la vitesse de transmission : de 50 à 2400 baud (bits 0 à 3)

Le VIC permet le full duplex, donc nous vous recommandons de le spécifier. En effet, lors de l'OPEN, le VIC réserve deux tampons de 256 octets en fin de mémoire, un pour l'émission, un pour la réception. Ceci a une conséquence très importante : TOUTE VARIABLE EST PERDUE APRES L'OPEN RS232. Celui-ci doit donc être, dans votre programme, avant toute utilisation d'une variable.

suivant le schéma :



6 5	longueur
0 0	8 bits
0 1	7 bits
1 0	6 bits
1 1	5 bits

vitesse

3 2 1 0	bauds
0 0 0 1	50
0 0 1 0	75
0 0 1 1	110
0 1 0 0	135
0 1 0 1	150
0 1 1 0	300
0 1 1 1	600
1 0 0 0	1200
1 0 0 1	1800
1 0 1 0	2400

Les combinaisons non citées correspondent à des vitesses inaccessibles au logiciel du VIC

La valeur W détermine :

. le mode d'interface : 3 lignes ou X lignes (bit 0)
. le mode duplex (bit 4)
. l'option sur la parité (bits 5 à 7) :

les options possibles sont :

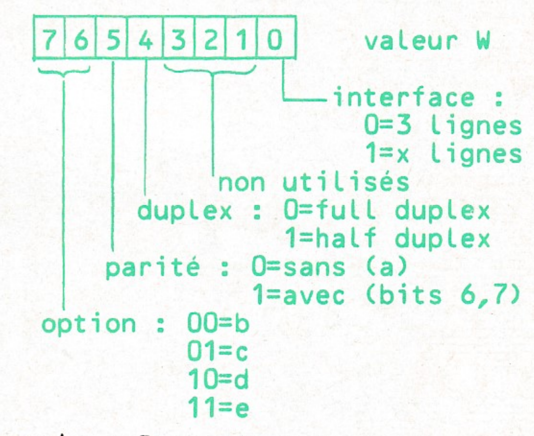
a) pas de bit de parité
b) parité impaire
c) parité paire
d) 1 envoyé systématiquement pas de contrôle à la réception
e) 0 envoyé systématiquement pas de contrôle à la réception.

L'option (d) est celle des télétypes. Elle se rencontre très souvent.

Les options de duplex sont :

. full duplex (émission et réception simultanées)
. half duplex (émission et réception alternées).

La valeur W obéit au schéma :



Exemple : Pour communiquer avec un télétype (7 bits, parité option (d), 2 bits d'arrêt, full duplex, interface 3 lignes, 110 bauds) l'OPEN sera :
OPEN 2,2,0, CHR$(163) + CHR$(160)

V = 163 = 128 (2 bits d'arrêt) + 32(7 bits) + 3(110 bauds)

W = 160 = 128 + 32 (parité d) + 0(full duplex) + 0(interface 3 lignes).

L'OPEN pour données 7 bits, parité absente, 1 bit d'arrêt, full duplex, interface 3 lignes, 300 bauds est :
OPEN 2,2,0, CHR$(38) + CHR$(0)

Transmission des données

L'écriture des données se fait, bien sûr, par PRINT#

Exemple : PRINT#2, A$;

La lecture doit se faire exclusivement caractère par caractère, par GET#.

Exemple : GET#2,A$

la fermeture se fait par CLOSE

Exemple : CLOSE 2.

Mais elle doit se faire uniquement si l'on est sûr qu'il n'y a plus de données à émettre, ce qui se teste par

IF(PEEK(37151)AND64)=0THENCLOSE2

La variable d'état ST

Comme d'habitude, si tout s'est bien passé ST = 0. Sinon, les valeurs suivantes ont une signification pour le RS232 :

ST	signification
1	erreur de parité
2	erreur de cadrage (bit d'arrêt manquant)
4	débordement du tampon-réception (l'émetteur envoie trop vite)
16	le signal CTS manque *
64	le signal DSR manque *
-128	détection de BREAK *

* n'existent pas en interface 3 fils.

Exemple complet

On trouvera ci-dessous un programme élémentaire qui gère le VIC comme un terminal RS232 : tout ce qui est lu au clavier est émis sur la ligne, tandis que tout ce qui est reçu est affiché sur l'écran.

Mise à part la limitation de l'écran à 22 colonnes, le VIC devient un terminal efficace, à très bas prix.

```
5 REM RS232 A 110 BD,7BITS+PAR=MARK
,2 STOP
10 OPEN2,2,2,CHR$(128+32+3)+CHR$(128
+32)
20 GET#2,A$
30 GET B$
40 IF B$<>"" THEN PRINT#2,B$;
50 GET#2,C$
60 PRINTC$;:GOTO30
```

Quelques précautions avec le RS232 du VIC

- Ne jamais ouvrir plus d'un fichier sur le périphérique 2(RS232) car la 2ème ouverture détruirait les tampons de la 1ère.

- LOAD et SAVE sont interdits sur le périphérique 2, mais CMD marche.

- pour employer une imprimante RS232, il faut disposer d'une imprimante ayant un double tampon d'entrée. En effet, lors d'un passage à la ligne, l'imprimante sera non prête pendant plus longtemps. Or, l'interface RS232 du VIC n'a aucune précaution pour cela, sauf si vous mettez des temporisations dans votre programme, ce qui est en tous cas impossible pour CMD suivi de LIST.

Daniel-Jean DAVID

Calcul de Factorielle N

Un exemple d'utilisation des routines de la ROM Basic.

L'intérêt de cet article réside surtout dans le fait qu'il donne le mode d'emploi des routines internes les plus fondamentales de l'interpréteur BASIC. Ceci vous permettra de les utiliser dans vos propres programmes.

1 - Introduction

Le programme BASIC du tableau 1 effectue le calcul de factorielle N% - (soit N%! = 1*2*3...*N%)- successivement selon 2 méthodes, et indique dans chaque cas, la durée du calcul en 1/60ème de sec.

La méthode 1 fait appel à un sous-programme en langage machine décrit ci-après. La méthode 2, purement BASIC, se passe de commentaires.

Les exemples de RUN montrent que les deux méthodes donnent le même résultat, ce qui constitue une vérification. La méthode 1 est sensiblement deux fois plus rapide (mais de toutes façons, c'est très rapide, et la différence est inappréciable).

NOTA : avec N%=34, on obtient 'OUT OF MEMORY ERROR'.

L'intérêt du programme réside dans l'exemple d'application du langage machine et d'utilisation des routines internes de l'interpréteur BASIC.

Les adresses des routines utilisées, selon les différentes versions Commodore, sont indiquées au tableau 5 : leur fonctionnement est détaillé en annexe.

2 - Description du programme langage machine

2-1 Principe.

On effectue successivement les opérations suivantes :

1 - Récupération de la variable Basic N%

2 - N% est placé en ACC1 par INTFLP, puis en ACC2 par ACC12

3 - On calcule N%=N%-1 et on place le résultat en ACC1, tant que N% XX 0; sinon on va en 5.

4 - On effectue N%*(N%-1) par MULT: résultat transféré dans ACC2 par ACC12, et on reboucle sur l'opération 3.

5 - On imprime le résultat par FLPASC et STROUT.

2-2 Analyse détaillée. (cf. Assembleur symbolique. Tableau 2.)

1030-1070 On fixe les adresses des routines (ici un CBM 4032)

1080 Début du programme en $0384 (soit 3*256+8*16+4=900 décimal, ce qui convient pour toutes versions CBM.

1090-1120 Voici la phase essentiel-

le de récupération de N% : les adresses 68 et 69 ($44 et 45) contiennent les octets bas et haut de l'adresse du 1er des 5 octets qui représentent la variable Basic utilisée en dernier lieu (ici N%).
Dans le cas d'une variable entière, le 1er octet contient le poids fort, le 2ème le poids faible, les 3 autres contiennent 0. Comme, dans notre cas, N% sera toujours 255, le 1er octet sera toujours 0: seul le 2ème nous intéresse. L'adressage indirect LDA(68),Y avec Y=1 récupère directement la valeur cherchée, sans qu'il soit besoin de savoir où se trouve rangée la variable N%. L'explication est longue, mais l'exécution, particulièrement concise, se réduit à: B1 44 !

Dans le cas du PET, les adresses 68 et 69 ($44 et 45) deviennent 150 et 151 ($96 et 97): l'instruction B1 44 doit alors être remplacée par B1 96 sur PET.

1130-1140 Poids fort dans (A)=0, poids faible dans (Y), INTFLP effectue le trasfert de N% dans ACC1.

1150 ACC12 le place dans ACC2.

1160-1180 N% est récupéré, mis dans (Y) et décrémenté.

1190 plutôt que d'aller jusque 0, on aurait pu s'arrêter à 1, par 'CPY =1' et 'BEQ RESULT.': on évite ainsi la dernière multiplication par 1, au prix de 2 octets en plus dans le programme.

1200-1280 on multiplie les nombres contenus dans ACC1 (N%) et ACC2 (N%-1). Le résultat contenu dans ACC1 est transféré dans ACC2.

1290-1300 'SEC BCS BOUCLE' est meilleur que 'JMP BOUCLE', car dans le premier cas, on revient en arrière d'un certain nombre d'octets, alors que dans le 2ème cas, on saute à une adresse absolue qui varie selon l'adresse de départ: le programme n'est plus aisément translatable.

1310-1360 'STROUT' écrit sur l'écran à la manière d'un 'PRINT' BASIC terminé par un ';', d'où la raison du 'PRINT' qui apparaît à la ligne 140 du programme BASIC (tableau 1) alors qu'il n'y en a pas à la ligne 280.

<u>REMARQUE</u>: les 'NOP' ($EA) qui parsèment le programme ont leur utilité: il est intéressant d'en remplacer certains par des 'BRK' ($00) qui rendent la main au moniteur L.M. ce qui permet d'examiner par exemple le contenu de ACC1 ($005E-0063) ou de ACC2 ($0066-006B) et de continuer en tapant simplement 'G' "RETURN".

<u>2-3 Assemblage et chargement.</u>

Le tableau d'assemblage 3 donne, en regard des adresses mémoires en Hexa et en décimal, les codes hexa du programme ainsi que les mnémoniques (avec une syntaxe quelque peu différente de celle du tableau 2) ainsi que les valeurs des opérandes en décimal, de $0384 à $03B5 (900 à 949 décimal).

Le tableau 4 donne la disposition en mémoire sous forme commode pour chargement direct par le moniteur L.M.

3 - Annexe : Fonctionnement des routines utilisées

<u>INTFLP</u>

Transforme en format flottant et transfère dans l'accumulateur numéro 1 (ACC1) le nombre entier compris entre -32768 ($8000) et +32767 ($7FFF) dont l'octet de poids fort a été placé dans (A) et celui de poids faible dans (Y).

Exemple : Avec un CBM 4000 : "LDA = 03 LDY =3A JSR $C4BC" place 826 dans ACC1 qui contient :

8A CE 80 00 00 00

<u>ACC12</u>

Transfère ACC1 dans ACC2.

Exemple : après "JSR $CD42" on retrouve dans ACC2 :

8A CE 80 00 00 00

MULT

Multiplie ACC1 par ACC2, résultat dans ACC1

Exemple : "JSR $CB66" donne ACC1 :

9E FF FF FF FF 20

FLPASC

Transforme en décimal le contenu de ACC1, et inscrit le résultat, en code ASCII, à partir de $0100, en terminant par 00.

Exemple : après "JSR $CF93" on trouve à partir de $0100 :

10 36 38 32 32 37 36 00

où 10 est le code "line feed": le reste est transparent : il suffit de ne prendre que les poids faibles, on aperçoit déjà le résultat 682276.

STROUT

Signifie "STRING OUT" : sortie de chaîne de caractères. Ecrit sur l'écran (plus exactement sur le périphérique de sortie en service) la chaîne de caractères terminée par 00 commençant à l'adresse dont le poids fort est dans (Y) et le poids faible dans (A). Attention ! c'est l'inverse du cas de INTFLP !

Exemple : Après "LDA =$00 LDY =$01 JSR $BB1D" on obtient sur l'écran 682276, qui est bien égal à: 826*826, C.Q.F.D. !

J. PIERRAT

TABLEAU 1 : PROGRAMME BASIC

```
10 REM J.PIERRAT VERSAILLES 20/08/82
20 REM CALCUL DE FACTORIEL N%
30 REM ======================
90 REM *** METHODE 1 ***
100 INPUT"N%=";N%
110 TI$="000000"
120 PRINT"N%!=";
130 SYS900
140 PRINT:PRINTTI
150 PRINT
200 REM *** METHODE 2 ***
220 TI$="000000"
230 F=1
240 FORI=1TON%
250 F=F*I
260 NEXT
270 PRINT"N%!=";F
280 PRINTTI
285 PRINT
290 GOTO100


RUN
N%=? 4
N%!= 24
 1

N%!= 24
 2

N%=? 10
N%!= 3628800
 2
```

TABLEAU 3 : ASSEMBLAGE

```
0384 900   A0 01      LDYIM 1
0386 902   B1 44      LDAIY 68
0388 904   A8         TAY
0389 905   48         PHA
038A 906   A9 00      LDAIM 0
038C 908   20 BC C4   JSR   50364
038F 911   20 42 CD   JSR   52546
0392 914   68         PLA
0393 915   A8         TAY
0394 916   88         DEY
0395 917   F0 13      BEQ   938
0397 919   98         TYA
0398 920   48         PHA
0399 921   A9 00      LDAIM 0
039B 923   EA         NOP
039C 924   20 BC C4   JSR   50364
039F 927   EA         NOP
03A0 928   20 66 CB   JSR   52070
03A3 931   20 42 CD   JSR   52546
03A6 934   EA         NOP
03A7 935   38         SEC
03A8 936   B0 E8      BCS   914
03AA 938   20 93 CF   JSR   53139
03AD 941   EA         NOP
03AE 942   A9 00      LDAIM 0
03B0 944   A0 01      LDYIM 1
03B2 946   20 1D BB   JSR   47901
03B5 949   60         RTS
```

TABLEAU 2 : ASSEMBLEUR SYMBOLIQUE

```
1000 ;VERSAILLES 19/08/82
1010 ;CALCUL DE FACTORIEL N%
1020 ;
1030        INTFLP=$C4BC
1040        ACC12=$CD42
1050        MULT=$CB66
1060        FLPASC=$CF93
1070        STROUT=$BB1D
1080        *=$0384          ;DECIMAL 900
1090        LDY #1           ;ON RECUPERE L'OCTET DE POIDS FAIBLE
1100        LDA (68),Y       ;DE N% DANS (A) ET (Y)
1110        TAY
1120        PHA              ;SAUVEGARDE N% DANS LA PILE
1130        LDA #0           ;POIDS FORT DE N% (=0) DANS (A)
1140        JSR INTFLP       ;(ACC1)←N%
1150        JSR ACC12        ;(ACC2)←(ACC1)
1160 BOUCLE PLA              ;RECUPERE N%
1170        TAY              ;DANS (Y)
1180        DEY              ;N%=N%-1
1190        BEQ RESULT       ;SI N%=0 RESULTAT
1200        TYA
1210        PHA              ;SAUVEGARDE N% DANS LA PILE
1220        LDA #0           ;REMET (A) A ZERO
1230        NOP
1240        JSR INTFLP       ;(ACC1)←N%
1250        NOP
1260        JSR MULT         ;(ACC1)←(ACC1)*(ACC2) SOIT (ACC1)←N%*(N%-1)
1270        JSR ACC12        ;(ACC2)←(ACC1)
1280        NOP
1290        SEC              ;FORCE LE JUMP
1300        BCS BOUCLE
1310 RESULT JSR FLPASC       ;(ACC1) EN ASCII A PARTIR DE $0100
1320        NOP
1330        LDA #0
1340        LDY #1
1350        JSR STROUT       ;IMPRIME RESULTAT SUR L'ECRAN
1360        RTS
```

TABLEAU 5 : ADRESSES DES ROUTINES UTILISEES

ROUTINES	PET	CBM3000	CBM8000 CBM4000
INTFLP	D278	D26D	C4BC
ACC12	DADE	DB18	CD42
MULT	D900	D93C	CB66
FLPASC	DCAF	DCE9	CF93
STROUT	CA27	CA1C	BB1D

TABLEAU 4 : PROGRAMME EN MEMOIRE

```
.M 0384 03B5
.:  0384 A0 01 B1 44 A8 48 A9 00
.:  038C 20 BC C4 20 42 CD 68 A8
.:  0394 88 F0 13 98 48 A9 00 EA
.:  039C 20 BC C4 EA 20 66 CB 20
.:  03A4 42 CD EA 38 B0 E8 20 93
.:  03AC CF EA A9 00 A0 01 20 1D
.:  03B4 BB 60 00 00 00 00 00 00
.X
```

Prêts Bancaires

Ce programme permet de traiter les différentes données d'un prêt, puis d'avoir le tableau d'amortissement (intérêt, amortissement, total par année légale) à l'écran ou sur imprimante.

Ce programme est très complet pour les calculs, le tableau d'amortissement est même plus complet que celui fourni par les banques puisqu'il donne les totaux annuels. Par contre, afin de le rendre rapide à taper au clavier, ce programme est très succinct pour le reste (voir remarques sur la programmation). Les bases des calculs ont été exposées dans le numéro 2 de la Commode.

Dans la première partie, s'il manque une quelconque des données du prêt (durée, capital emprunté, taux d'intérêt actuariel ou nominal, montant de chaque remboursement) le programme calcule la donnée manquante. Ceci peut être utilisé par exemple :

- pour calculer le montant de chaque remboursement d'un prèt donné.

- pour connaitre le prêt maximum possible sans que les remboursements dépassent 25 ou 30 % des revenus (condition souvent exigée par les prêteurs).

- pour comparer plusieurs prêts entre eux en calculant pour chacun d'eux un intérêt réel qui tienne compte des frais de dossier et d'hypothèque parfois très différents d'un prêt à un autre.

- pour savoir si le préteur a vraiment indiqué l'intérêt actuariel et non pas le nominal (nettement plus faible).

A ce sujet, rappelons que lorsqu'il y a plusieurs versements annuels, il faut distinguer intérêt actuariel (ou effectif) et intérêt nominal. Le nominal ne tient pas compte de la composition de l'intérêt à chaque versement alors que l'actuariel en tient compte. Par exemple, pour des remboursements mensuels avec un intérêt nominal de 20 % , l'intérêt actuariel est de 20,93%. Les (bonnes) banques françaises indiquent en principe l'actuariel mais vérifiez-le par ce programme en faisant recalculer l'intérêt annoncé (il faut pour cela connaître durée, capital, remboursements).

La deuxième partie du programme fournit le tableau d'amortissement. Les résultats affichés indiquent pour chaque remboursement le numéro du mois, la part intérêt, la part amortissement et le capital restant dû.

A la fin de chaque année légale, le programme affiche le total annuel des intérêts et des amortissements. Ceci est très utile sur le plan comptable et fiscal. Par exemple, les intérêts des prêts professionnels et pour l'habitation principale étant déductibles des revenus, il peut être utile pour choisir entre plusieurs prêts possibles de savoir lequel donne le maximum de déduction les premières années, ou si l'on dépassera et pendant combien d'années le maximum de déduc-

tion autorisée pour l'habitation.

Ce programme est très simple à utiliser pour des prêts à intérèt fixe pendant toute la durée du prèt et à remboursements constants (cas le plus fréquent).

Mais il peut être également utilisé (de façon plus compliquée) pour des taux d'intérêt variables et/ou des remboursements variables. Par exemple, si l'on désire avoir le tableau d'amortissement pour un prêt qui a un taux d'intérêt I1 et des remboursements R1 pendant 5 ans puis I2 et R2 pendant 10 ans : fournir I1, R1 et le capital emprunté et ne pas donner la durée; prendre les 5 premières années du tableau d'amortissement ; utiliser le capital restant dû à la fin de la 5ème année pour faire comme s'il s'agissait d'un nouveau prêt commençant dans 6 ans avec un intérèt I2 et des remboursements R2.

Le CBM n'utilise qu'un peu plus de 9 chiffres significatifs pour faire les calculs. Aussi pour des prêts de plusieurs centaines de milliers de francs (nouveaux) sur 20 ou 30 ans, on a à la fin du tableau des erreurs de quelques centimes par rapport au même tableau fourni par le gros ordinateurs des banques (l'auteur a fait cette comparaison pour plusieurs prêts). En particulier, dans la colonne capital restant dû, lors du dernier remboursement, il peut y avoir quelques centimes en plus ou en moins. Pour des durées d'une centaine d'années l'imprécision peut devenir plus grave.

Lorsque dans la première partie on a fait calculer la durée du prêt il faudrait un hasard étonnant pour tomber sur une durée correspondant à un nombre entier de remboursements. Aussi le programme arrête le tableau d'amortissement au dernier remboursement normal et affiche le solde à rembourser (ligne 690).

Remarques sur la programmation

Ce programme a été fait le plus court possible afin de faciliter la tâche de ceux qui voudront le taper. Aussi, il ne comporte pas de protection des inputs et aucun contrôle de la validité des données fournies (si on propose de rembourser un prêt avec des remboursements qui ne payent même pas les intérêts, le programme se lance quand même dans les calculs). Il faut donc penser à fournir des montants de remboursements plausibles quand on fait calculer la durée ou le taux d'intérêt. Sinon on sortira du programme sur un message d'erreur (division by zéro, par exemple) ou on obtiendra des résultats aberrants.

La présentation à l'écran n'aligne pas les nombres à droite. Celle sur papier est zigzagante mais lisible (l'option imprimante n'occupe que 4 lignes de programme).

Ceux qui ne veulent pas imprimer peuvent supprimer les lignes 550, 560, 660, 700 en remplaçant X par 28 aux lignes 500 et 610.

On peut supprimer tous les REM sans inconvénients. Si l'on ne désire pas le tableau d'amortissement, supprimer les lignes 500 à 740.

Programmation des données du prêt (210 à 430)

Lignes 340 à 370 : calcul de l'intérêt actuariel (IA) à partir de l'intérêt nominal (I), ou inversement. La ligne 370 forme un petit sous-programme car elle est également utilisée en 930.

Lignes 380 et 400 à 420 : calcul de la donnée qui n'a pas été fournie par l'utilisateur : durée (D), capital (C), intérêt (I) ou remboursement (R). Envoi à 4 sous-programmes placés de 820 à 960. Si l'utilisateur a fourni les 3 premières données, le programme (ligne 380) refuse l'entrée de la 4ème(R). En effet, la valeur des remboursements fournie par les banques est arrondie au centime près. Or, pour faire le tableau d'amortissement, il faut connaître (R) avec le maximum de précision. Le programme fait ce calcul et affiche le

résultat arrondi.

Le calcul de l'intérêt (890 à 930) se fait par approximations successives car il n'existe pas de formule directe. Les formules ont été choisies pour essayer d'éviter les overflows et les boucles sans fin. Cependant, ceci peut se produire pour des valeurs considérables (milliards de francs ou centaines d'années de durée).

Programmation du tableau d'amortissement (510 à 740)

En fournissant la date du premier remboursement (510 à 530) on permet, entre autres, de faire les totaux par années légales (actuellement, il faut toujours penser au fisc).

Le même programme d'affichage sert à la fois pour l'écran et l'imprimante. Mais les ordres TAB sont interprétés par les imprimantes 3022 et 4022 comme des ordres SPC. Ceci explique l'aspect zigzagant de l'impression. On a introduit la variable X (lignes 560, 580, 610) pour adapter un peu l'affichage à l'imprimante. Pour bien faire, il faudrait écrire un programme d'impression différent de celui d'affichage à l'écran, mais ceci allongerait nettement le programme.

Les variables introduites dans cette 2ème partie et qui sont utilisées de façon répétée sont :

H = Date de chaque remboursement. R% = nombre de remboursements à faire. RE =nombre de remboursements déjà effectués. CD = capital restant dû. IR et A = intérêt et amortissement totaux de chaque année. L = nombre de lignes affichées sur l'écran.

Les calculs se font par une boucle de 570 à 680 à l'intérieur de laquelle ont lieu plusieurs tests pour savoir entre autres s'il faut afficher le total annuel (620) et si l'écran va être plein ou si c'est terminé (650).

Alain TASSY

```
100 PRINT"■                     ■PRETS BANCAIRES■
110 PRINT"■■■     AUTEUR : ALAIN TASSY (AOUT 1982)
120 PRINT"■LA RESERVE. RUE CONDORCET. 83000 TOULON.
130 PRINT"■■■    POUR COMMODORE SERIE 3000 ET 4000.
140 PRINT"■■■■■■■■■TAPEZ TOUCHE SHIFT":WAIT 152,1
150 :
200 REM===1ER PARTIE : LES DONNEES DU PRET.
210 PRINT"■      ■VOUS ALLEZ DEVOIR ME DONNER■ LE NOM-
215 PRINT"■BRE ANNUEL DE REMBOURSEMENTS ET 3 DES 4
220 PRINT"■VALEURS SUIVANTES:
230 PRINT"■- DUREE DU PRET
240 PRINT"- CAPITAL EMPRUNTE
245 PRINT"- TAUX D'INTERET ANNUEL (ACTUARIEL OU
250 PRINT"  NOMINAL. AU CHOIX)
255 PRINT"- MONTANT DE CHAQUE REMBOURSEMENT
260 PRINT"■   POUR LA VALEUR QUE VOUS NE ME DONNE-
270 PRINT"■REZ PAS VOUS TAPEREZ 0 (0,0 SI C'EST LA
275 PRINT"■DUREE).
280 INPUT"■NOMBRE ANNUEL DE REMBOURSEMENTS ";N
290 PRINT"■DUREE DU PRET SOUS LA FORME: ANNEE,MOIS
300 INPUT"(EX. POUR 15 ANS JUSTE : 15,0) ";DA,M:D=(12*DA+M)/12
310 INPUT"■CAPITAL EMPRUNTE ";C
320 INPUT"■TAUX D'INTERET ANNUEL (EN %) ";I:I=I/100:IF N=1 OR I=0 GOTO 380
330 PRINT"EST-CE UN TAUX ACTUARIEL OU NOMINAL
340 INPUT"(TAPEZ A OU N) ";I$:IF I$<>"A"AND I$<>"N"THEN 340
350 IF I$="A"THEN I=((I+1)↑(1/N)-1)*N:PRINT"INTERET NOMINAL ="I*100:GOTO 380
360 IF I$="N"THEN GOSUB 370:GOTO 380
370 IA=((I/N+1)↑N)-1:PRINT"■INTERET ACTUARIEL ="IA*100:RETURN
380 IF C>0 AND D>0 AND I>0 THEN PRINT:GOSUB 950:GOTO 430
```

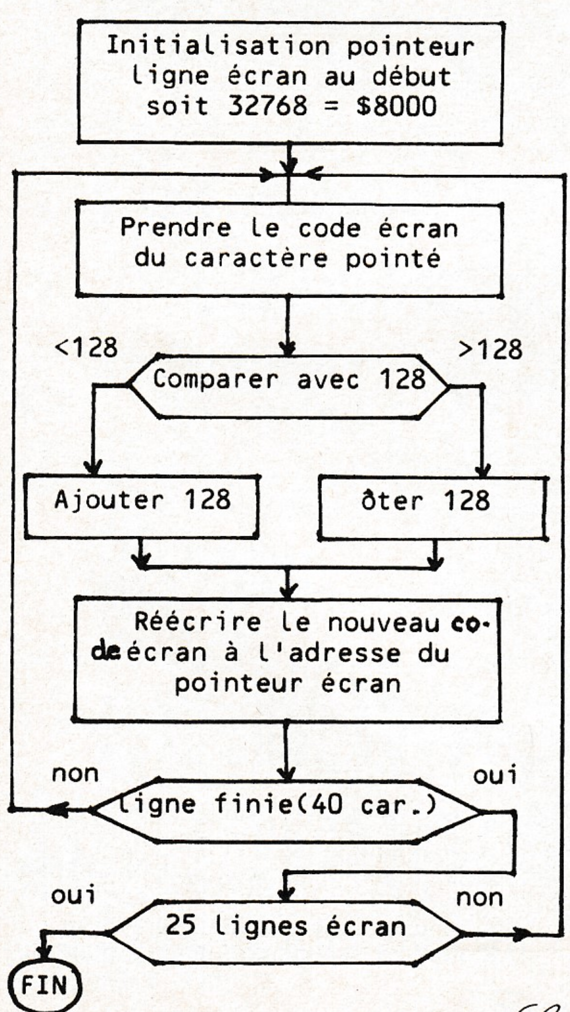
```
390 INPUT"CHAQUE REMBOURSEMENT = ";R:PRINT
400 IF D=0 THEN GOSUB 820
410 IF C=0 THEN GOSUB 860
420 IF I=0 THEN GOSUB 890
430 INPUT"TABLEAU AMORTIS: 1. AUTRE CALCUL: 2 ";Z:ON Z GOTO 510,210
440 :
500 REM===2EM PARTIE : TABLEAU D'AMORTISSEMENT
510 PRINT"DATE DU PREMIER REMBOURSEMENT SOUS LA
520 PRINT"FORME: MOIS,ANNEE (EX. POUR MARS 1982
530 INPUT"TAPER 3,1982) ";M,P
540 H=P+M/12-.08:R%=D*N:CD=C:IA=0:AA=0:L=0
550 INPUT"FAUT-IL IMPRIMER LE TABLEAU (O/N)";Z$
560 X=28:IF Z$="O"THEN X=16:OPEN 4,4:CMD 4
570 FOR RE=1 TO R%:IF L>0 GOTO 590
580 PRINT"MOIS INTERETS" TAB(17)"AMORTIS." TAB(X)"CAP.REST.DU":PRINT:L=2
590 IR=CD*I/N:A=R-IR:CD=CD-A:IA=IA+IR:AA=AA+A
600 PRINT INT((H-INT(H))*12+1) TAB(4)INT(IR*100+.5)/100;
610 PRINT TAB(16)INT(A*100+.5)/100 TAB(X)STR$(INT(CD*100+.5)/100):L=L+1
620 IF H+1/N<INT(H+1) AND RE<R% GOTO 680
630 PRINT"TOTAL ANNEE"INT(H)
640 PRINT TAB(4)INT(IA*100+.5)/100 TAB(16)INT(AA*100+.5)/100:PRINT
650 IA=0:AA=0:L=L+3:IF L+N+3<25 OR RE=R% GOTO 680
660 IF Z$="O" GOTO 680
670 PRINT"POUR CONTINUER TAPER TOUCHE SHIFT":L=0:WAIT 152,1
680 H=H+1/N:NEXT
690 IF D*N>R% THEN PRINT"      RESTE A REMBOURSER :"INT(CD*100+.5)/100
700 IF Z$="O"THEN PRINT# 4:CLOSE 4:PRINT
710 PRINT"REVOIR AMORTISS. 1. AUTRES CALCULS 2
720 GET A$:IF A$="1"THEN PRINT"":GOTO 540
730 IF A$="2"GOTO 210
740 GOTO 720
790 :
800 REM======SOUS-PRG. DE CALCULS
810 REM===CALCUL DUREE DU PRET
820 D=-(LOG(1-(C*I)/(N*R))/(LOG(1+I/N)*N))
830 M=(D-INT(D))*12:J=(M-INT(M))*30
840 PRINT"DUREE DU PRET ="INT(D)"ANS"INT(M)"MOIS"INT(J)"JOURS":RETURN
850 REM===CALCUL CAPITAL EMPRUNTE
860 C=R*N*(1-1/(I/N+1)↑(N*D))/I
870 PRINT"CAPITAL EMPRUNTE ="INT(C*100+.5)/100:RETURN
880 REM===CALCUL INTERET (NOMINAL)
890 PRINT"JE CALCULE":I=.2:I1=0
900 RP=(I*C/N)/(1-1/((I/N+1)↑(N*D))):I2=ABS(I-I1):I1=I
910 IF RP>R+.001 THEN I=I-I2/2:GOTO 900
920 IF RP<R-.001 THEN I=I+I2:GOTO900
930 GOSUB 370:PRINT"INTERET NOMINAL ="I*100:RETURN
940 REM===CALCUL MONTANT DE CHAQUE REMBOURSEMENT
950 R=(I*C/N)/(1-1/(I/N+1)↑(N*D))
960 PRINT"CHAQUE REMBOURSEMENT ="INT(R*100+.5)/100:RETURN
```

Le programme TREVERSE

Etant donné une image-écran, voici un programme qui inverse la "vidéo": ce qui est vert sur fond noir devient noir sur fond vert et vice-versa.

En écrivant un tel programme en BASIC, il faudrait faire des POKE dans la mémoire écran, et ce sur une boucle variant de 1 à 1000. L'éxécution de ce programme ne donnerait pas un effet saisissant d'inversion vidéo. Une seule solution, écrire ce programme en langage machine.

Logigramme du sous-programme en langage machine



Quelques adresses

* Le début de la mémoire écran est en 32768 soit $8000 en hexadécimal.

* Le pointeur de ligne écran est situé aux adresses hexadécimales $C4 et $C5

$C4 : partie haute de l'adresse pointée
$C5 : partie basse de l'adresse pointée.

* Ce sous-programme sera chargé dans le tampon du magnétophone cassette numéro 2 dont l'adresse de départ est 826 soit $033A.

Listing du sous-programme démonstration

Ci-après 2 listings

- le listing du sous-programme entré dans le CBM à l'aide d'un assembleur (Assembleur RAC/ASSEM-02 de la Règle à calcul")

- le listing du programme BASIC permettant de charger ce sous-programme avec une démonstration écran.

Parties du programme

Dans le programme TREVERSE

. lignes 20 à 23 : chargement du sous-programme en langage machine à partir de l'adresse 826 = $033A

. lignes 27 à 31 : sous-programme en langage machine (valeurs obtenues par l'assemblage).

. lignes 33 à 42 : Programme BASIC de démonstration.

En décalant l'écran vers le haut par SYS 58687. cela permet d'écrire 3 fois la même image sans se préoccuper des lignes écran.

Les lignes 22 et 23 permettent de convertir simplement de l'hexadécimal en décimal.

Ce programme n'est donné qu'à titre d'exemple. L'essentiel est le sous-programme d'inversion vidéo. Il s'interrompt par appui sur la touche RUN/STOP.

A. HESBOIS

```
10 "033A        *                :*******************************************
11 "033A        *                :S/P D'INVERSION VIDEO
12 "033A        *                :COPYRIGHT A.HESBOIS MAI 82
13 "033A        *
14 "033A        *       *=$033A  :ORIGINE S/P
15 "C400        *       BE=$C4   :PARTIE BASSE ADR. ECRAN
16 "C500        *       HE=$C5   :PARTIE HAUTE ADR. ECRAN
17 "033A        *
18 "033A A200   *       LDX#0    :0->POINTEUR X
19 "033C A900   *       LDA#0    :0->ACCUMULATEUR (REPERE ACC)
20 "033E 85C4   *       STA BE   :INITIALISATION POINTEUR ECRAN
21 "0340 A980   *       LDA#$80  :             AVEC
22 "0342 85C5   *       STA HE   :ECRITURE DE $8000 EN $C4,$C5
23 "0344        *
24 "0344 A000   *BOUCP  LDY#0    :0->POINTEUR Y
25 "0346 B1C4   *INVL   LDA(BE),Y :[ADRESSE] POINTEE PAR BE ->ACC
26 "0348 C980   *       CMP#128  :COMPARAISON DE [ACC] AVEC 128
27 "034A 1005   *       BPL PLUS :SI >=128 BRANCHEMENT EN PLUS
28 "034C 6980   *       ADC#128  :[ACC]+128
29 "034E 4C5303*       JMP SUIT1 :SAUT EN SUIT1
30 "0351 E980   *PLUS   SBC#128  :[ACC]-128
31 "0353 91C4   *SUIT1  STA(BE),Y :[ACC]->ADR POINTE PAR BE
32 "0355 C8     *       INY      :[Y]+1
33 "0356 C028   *       CPY#40   :COMPARE [Y] AVEC 40 (40 CARACTERES/LIGNE)
34 "0358 D0EC   *       BNE INVL :SI [Y]<40 REBRANCHEMENT EN INVL
35 "035A 18     *       CLC      :RAZ RETENUE
36 "035B A928   *       LDA#40   :[ACC]+40
37 "035D 65C4   *       ADC BE   :ON AJOUTE DONC 40
38 "035F 85C4   *       STA BE   :AU POINTEUR ECRAN
39 "0361 9002   *       BCC SUIT2 :PAS DE REPORT ADDITION SAUT EN SUIT2
40 "0363 E6C5   *       INC HE   :REPORT -> PARTIE HAUTE ADR.POINT.ECRAN +1
41 "0365 E8     *SUIT2  INX      :[X]+1
42 "0366 E019   *       CPX#25   :COMPARE [X] AVEC 25 (25 LIGNES ECRAN)
43 "0368 D0DA   *       BNE BOUCP :SI [X]<25 REBRANCHEMENT EN BOUCP
44 "036A D8     *       CLD      :MISE A ZERO DU MODE DECIMAL
45 "036B 60     *       RTS      :FIN DE S/P -> RETOUR
46 "036C        *
47 "036C*       *       .END     :FIN D'ASSEMBLAGE
```

```
11 REM*************************
12 REM*  PROGRAMME TREVERSE  *
13 REM*    DEMONSTRATION      *
14 REM*  VIDEO     INVERSEE  *
15 REM* COPYRIGHT A. HESBOIS *
16 REM*       MAI 1982       *
17 REM*************************
18 :
19 REM CGHT S/P LANGAGE MACHINE
20 READ L
21 READ A$:IFA$="*" THEN 34
22 A=ASC(A$)-48:B=ASC(RIGHT$(A$,1))-48:C=LEN(A$)
23 N=B+7*(B>9)-(C=2)*(16*(A+7*(A>9))):POKEL,N:L=L+1:GOTO21
24 :
25 REM S/P LANGAGE MACHINE
26 DATA 826
27 DATA A2,00,A9,00,85,C4,A9,80,85,C5
28 DATA A0,00,B1,C4,C9,80,10,05,69,80
29 DATA 4C,53,03,E9,80,91,C4,C8,C0,28
30 DATA D0,EC,18,A9,28,65,C4,85,C4
31 DATA 90,02,E6,C5,E8,E0,19,D0,DA,D8,60,*
32 :
33 REM  PROGRAMME DEMONSTRATION
34 PRINT"□"
35 FORI=1TO3:SYS58687:PRINTTAB(8)"▒****************************************
36 PRINTTAB(8)"▒*                   *
37 PRINTTAB(8)"▒* VIDEO   INVERSEE *
38 PRINTTAB(8)"▒*  SUR  CBM  3032  *
39 PRINTTAB(8)"▒*                   *
40 PRINTTAB(8)"▒*********************
41 SYS58687:NEXT
42 FORI=1TO900:NEXT:SYS826:GOTO42
```

Les Cassettes LA COMMODE

La Commode se lance dans la commercialisation de cassettes

Pour la premiére cassette, on frappe un grand coup car il s'agit de UTIL-VIC de H. LE MARCHAND, un programme qui ajoute des fonctions au BASIC de votre VIC :

SET X,Y	établit les couleurs écran et cadre
CLEAN	vide l'écran
PLOT X,Y	trace un point en haute résolution
JOYO	lit les joysticks
LOOK /chaine/	recherche une chaine
KEY X "chaine"	affecte une fonction à une touche de fonction
PAUSE	permet de suspendre un listing par shift
etc...	

En plus, chaque mot clé a une abréviation obtenue par SHIFT et une lettre (Exemple : SHIFT P = POKE etc...)

BON DE COMMANDE

NOM :
ADRESSE :

commande une cassette UTIL-VIC   version 3K □   version 8K ou + □
prix 60 F TTC

COMMUNIQUEZ

AVEC

TRANSNEX

pour CBM 4000/8000

- Transnex asynchrone : carte d'interface RS232C/V24
  connectée sur bus extension mémoire
  + extension Basic communications.

- Commodial : logiciel d'accès banques de données (QUESTEL...)
  services de messagerie électronique (MISSIVE...)
  livré avec programme source.

En option : adaptateur MINITEL permettant de relier un CBM équipé de TRANSNEX à un terminal MINITEL.

En option : adaptation CBM 3000 / 4000 petit écran.

DOMAINES D'UTILISATION:

- Accès banques de données sur réseau téléphonique ou TRANSPAC (jusqu'à 1200 bauds full duplex)
- Services de messages électroniques
- Transfert de fichiers entre CBM
- Transfert de fichiers de texte TRAITEXT vers télécomposeuse (logiciel COMPODIAL)
- Liaison à distance avec un autre ordinateur
- Liaisons locales
- Interfaçage de périphériques spéciaux V24 ou boucle de courant (multipoints) (crayon-lecteur, horloges pointeuses, balances de précision etc...)



19-21, RUE MATHURIN-RÉGNIER - 75015 PARIS
TÉL.: 306.82.02 - TÉLEX: 204 875 F.

Ajustement de courbes

I - Présentation

1) Dans le cadre de l'étude d'un événement pouvant se traduire sous forme d'une fonction Y = f(X); si l'on trace le graphique de ce phénomène, on peut observer qu'il suit une loi de type :

Linéaire
Exponentielle
Logarithmique
Puissance.

Ce programme nous permettra de calculer la valeur des coefficients déterminant l'équation de la courbe ajustant au mieux vos résultats expérimentaux.

2) La méthode utilisée est celle des moindres carrés, le phénomène observé comportant "n" résultats, on peut avoir un tableau

x(1), x(2), x(3) ... x(n)
y(1), y(2), y(3) ... y(n)

qui forme un ensemble de couples (x(i), y(i)).

On calcule les quantités:

$$X = \sum_{i=1}^{n} x(i) \quad \text{somme des x}$$

$$Y = \sum_{i=1}^{n} y(i) \quad \text{somme des y}$$

$$XY = \sum_{i=1}^{n} x(i)*y(i) \quad \text{somme des produits xy}$$

$$X2 = \sum_{i=1}^{n} x(i)^2 \quad \text{somme des carrés des x}$$

$$Y2 = \sum_{i=1}^{n} y(i)^2 \quad \text{somme des carrés des y}$$

D'après la méthode des moindres carrés, sachant que le phénomène est traduit sous forme d'une équation du type Y = BX + A, on détermine les coefficients A et B de la façon suivante :

$$B = \frac{XY - (Y * X) / N}{X2 - (X * X) / N}$$

$$A = \frac{Y - (B * X)}{N}$$

Ensuite, on détermine le coefficient de corrélation qui mesure la fiabilité des coefficients A et B; on le note RZ et, plus il est proche de 1, plus les coefficients sont fiables.

$$RZ = \sqrt{\frac{(A*Y2) + (B*XY) - ((Y*Y) / N)}{Y2 - (Y*Y) / N}}$$

3) Suivant le choix de l'utilisateur, on détermine les équations suivantes.

<u>Type de loi</u>	<u>équation</u>
Linéaire	$Y = A + B.X$
Exponentielle	$Y = A\ e^{B.X}$
Logarithmique	$Y = A + B \text{ Log } X$
Puissance	$Y = A\ X^{B}$

II - Caractéristiques du programme

Langage : Basic.
Place mémoire 8 K.
Ecrit pour CBM 2000/3000/4000.

III - Organisation du programme

Lignes 0 à 190: présentation et entrée du type de courbe.

Lignes 190 à 350: calculs (cf. ci-dessus).

Lignes 500 à 600: résultats et affichage régression linéaire.

Lignes 600 à 680: affichage régression exponentielle.

Lignes 700 à 780: affichage régression logarithmique.

Lignes 800 à 880: affichage régression puissance.

Lignes 900 à 950: un autre calcul ?

Lignes 10000 à 10100(sous prg) aiguillage suivant choix du type de courbe avec confirmation du choix.

Lignes 20000 à 20210(sous prg) entrée des couples de valeurs X, Y avec possibilité de correction si erreur.

J. BRUGEASSOU

```
0 POKE59468,14:POKE59490,60:REM"■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■
1 REM"■■■■■■    ♠JUSTEMENT DE COURBES
5 PRINT"■■■■■■"
6 REM GOSUB30000:REM  EXPLIQUATION
10 REM BRUGEASSOU JACQUES LE 25/03/81
20 REM ------------------------------------
30 REM  PLACE MEMOIRE  4609 OCTETS
31 REM ------------------------------------
32 :
37 PRINT"■": GOSUB30000:PRINT"■"
40 PRINT"■■   ♠JUSTEMENT DE COURBES
41 L$="■■╮╱▔♠╮▁▔"
42 E$="■▔♠┳╱▔╱│╮▔┗┻▔"
43 C$="■┗▔│♠▁ ║ │╲╮●,▔"
44 P$="■▔╮♠♠♠♠╱─▔"
50 PRINT
60 PRINT"  ♠U'EL TYPE DE COURBES ..?
70 PRINT:PRINT
80 PRINT"       COURBE ■■INEAIRE
90 PRINT
100 PRINT"       COURBE ■■XPONENTIELLE
110 PRINT
120 PRINT"       COURBE L■■GARYTHMIQUE
130 PRINT
140 PRINT"       COURBE ■■UISSANCE
150 PRINT
160 GOSUB10000
170 PRINT"■":POKE59468,12:REM"■■■■■■■■■■■■■■■■■■
180 GOSUB20000
190 REM  ---------------
191 REM  --   CALCULS  --
192 REM  ---------------
195 FORI=1TON
200 X=X(I)+X             :REM  SOMME X
210 Y=Y(I)+Y             :REM  SOMME Y
220 XY=X(I)*Y(I)+XY      :REM  SOMME X*Y
230 X2=X(I)*X(I)+X2      :REM  SOMME X↑2
240 Y2=Y(I)*Y(I)+Y2      :REM  SOMME Y↑2
250 NEXTI
260 REM -- CALCUL DE B --
270 P=XY-((Y*X)/N)
280 PP=X2-((X*X)/N)
290 B=INT(100*P/PP)/100
```

```
295 REM -- CALCUL DE A --
300 P=Y-(B*X)
310 A=INT(100*P/N)/100
320 REM -- CALCUL DE R2 --
330 P=XY-(X*Y/N):P1=(X2-(X*X/N))*(Y2-(Y*Y/N))
340 P1=SQR(P1)
350 R2=ABS(INT(100*P/P1)/100)
395 REM ----------------
400 REM --  RESULTATS  --
410 REM ----------------
420 ONCOGOTO500,600,700,800
500 PRINT"███"
510 PRINT"------------------------------------------------"
520 PRINT"██          █REGRESSION LINEAIRE█
530 PRINT"██ LA COURBE EST DE LA FORME Y=A+B*X
535 PRINT
540 PRINT"EQUATION DE LA COURBE
550 PRINT"████████████████████":PRINT
560 PRINT"   Y=";A;"+";B;"*X
570 PRINT"██COEF DE REGRESSION R2=";R2:PRINT
580 GOTO900
600 PRINT"███"
605 A=INT(100*EXP(A))/100
610 PRINT"------------------------------------------------"
620 PRINT"██       █REGRESSION EXPONENTIELLE█
630 PRINT"██LA COURBE EST DE LA FORME Y=A*E↑(B*X)
635 PRINT
640 PRINT"EQUATION DE LA COURBE
650 PRINT"████████████████████":PRINT
660 PRINT"   Y=";A;"*EXP(";B;"*X)
670 PRINT"██COEF DE REGRESSION R2=";R2:PRINT
680 GOTO900
700 PRINT"███"
710 PRINT"------------------------------------------------"
720 PRINT"██       █REGRESSION LOGARYTHMIQUE█
730 PRINT"██LA COURBE EST DE LA FORME Y=A+B*LOG(X)
735 PRINT
740 PRINT"EQUATION DE LA COURBE
750 PRINT"████████████████████":PRINT
760 PRINT"   Y=";A;"+";B;"*LOG(X)
770 PRINT"██COEF DE REGRESSION R2=";R2:PRINT
780 GOTO900
800 PRINT"███"
805 A=INT(EXP(A)*100)/100
810 PRINT"------------------------------------------------"
820 PRINT"██          █REGRESSION PUISANCE█
830 PRINT"██LA COURBE EST DE LA FORME Y=A*X↑B
835 PRINT
840 PRINT"EQUATION DE LA COURBE
850 PRINT"████████████████████":PRINT
860 PRINT"   Y=";A;"*X↑";B;"
870 PRINT"██COEF DE REGRESSION R2=";R2:PRINT
880 GOTO900
900 PRINT"________________________________________________"
910 PRINT"█          UNE AUTRE ETUDE █O█UI  █N█ON      "
920 PRINT"________________________________________________"
930 GETRP$:IFRP$=""THEN930
940 IFRP$="O"THENCLR:POKE59468,14:GOTO40
950 PRINT"█":END
9999 END:REM"██████████"
```

```
10000 PRINT"CODE S.V.P ";"";
10010 GETCO$:IFCO$=""THEN10000
10020 IFCO$="L"THENPRINTL$:CO=1:GOTO10060
10030 IFCO$="E"THENPRINTE$:CO=2:GOTO10060
10040 IFCO$="O"THENPRINTO$:CO=3:GOTO10060
10050 IFCO$="P"THENPRINTP$:CO=4:GOTO10060
10055 RUN
10060 PRINT"  EXACT  OUI   NON  ";
10070 GETRP$:IFRP$=""THEN10070
10080 IFRP$="N"THENPRINT"NON":FORI=1TO1000:NEXT: GOTO40
10090 PRINT"OUI":FORI=1TO1000:NEXT
10100 RETURN
19999 REM -------------------------
20000 REM *SBR  ENTREE DES DONNEES*
20001 REM -------------------------
20005 PRINT"POUR UNE CORRECTION TAPEZ @
20010 PRINT"  NOMBRE DE POINTS  ";
20020 GETNO$:IFNO$=""THEN20020
20030 N$=N$+NO$:IFASC(NO$)<>13THEN20020
20040 N=VAL(N$):PRINTN
20045 DIMX(N),Y(N)
20050 FORI=1TON
20060 L$="                                  "
20065 C$="                                  "
20070 PRINT"VALEUR DE X(";I;")=";
20075 INPUTX$:IFX$="@"THENGOTO40000
20080 X(I)=VAL(X$)
20090 IFCO=1THEN20130
20100 IFCO=2THEN20130
20110 IFCO=3THENX(I)=LOG(X(I)):GOTO20130
20120 IFCO=4THENX(I)=LOG(X(I))
20130 PRINT"VALEUR DE Y(";I;")=";
20135 INPUTY$:IFY$="@"THENGOTO40000
20140 Y(I)=VAL(Y$)
20150 IFCO=1THEN20180
20160 IFCO=3THEN20180
20170 Y(I)=LOG(Y(I))
20180 PRINTL$
20190 PRINTC$
20200 NEXTI
20210 RETURN
30000 REM  ------------------
30040 FORI=1TO10:PRINT"----";:NEXT
30050 PRINT"
30060 PRINT"
30065 PRINT
30070 PRINT"
30080 PRINT
30090 PRINT" N FONCTION DE POINTS EXPERIMENTAUX CE
30100 PRINT" PROGRAMME VOUS CALCULERA LA COURBE
30110 PRINT" OBTIMUM ...
30120 PRINT"
30130 PRINT"LE FACTEUR DE REGRESSION VOUS FOURNIRA
30140 PRINT"LA PRECISION SUR LE RESULTAT
30145 FORI=1TO10:PRINT"----";:NEXT
30150 FORI=1TO10:PRINT"____";:NEXT
30160 PRINT"        APPUYER SUR UNE TOUCHE         "
30165 FORI=1TO10:PRINT"----";:NEXT
30170 GETA$:IFA$=""THEN30170
30180 RETURN
```

```
40000 REM -- CORRECTION --
40010 PRINT"POUR QUEL POINT";
40020 INPUTCR
40030 IFCR>NTHEN40010
40040 INPUT"NOUVELLE VALEUR POUR X ";X(CR)
40050 INPUT"NOUVELLE VALEUR POUR Y ";Y(CR)
40060 PRINT""
40070 GOTO20060
```

Bourse d'échange des programmes

Voici 6 programmes proposés par Claude BALZANO - 33 rte de Dourdan - 91520 EGLY:

Musique

CARACTERISTIQUES TECHNIQUES:

Ce programme fonctionne sur un CBM 4000, et occupe 5700 octets.
Il nécessite une interface sonore, du genre décrit dans La Commode no.1 par exemple.

FONCTION:

Transforme le clavier du CBM en clavier musical, du genre piano.
La note jouée est représentée sur la portée, et son nom est indiqué, si on le désire.

PRECISIONS COMPLEMENTAIRES:

1 - Le programme donne à choisir un timbre parmi quatre, qu'il faut écouter.
2 - Etendue musicale: un peu plus d'un octave. Demi-tons comme sur un piano.
3 - Le fait d'enclencher la touche SHIFT supprime l'impression du nom de la note (enseignement de la musique).
4 - Listing structuré et commenté.

Changement de base de numération

CARACTERISTIQUES TECHNIQUES:

Fonctionne sur un CBM 4000, et occupe 5200 octets. Il est enregistré sur cassette.

FONCTION:

Conversion d'entiers d'une base de numération dans une autre.

Les bases acceptées vont de la base 2 à la base 36 incluses.

PRESENTATION:

1 - Affichage des résultats dans un tableau à 2 colonnes, une pour chaque base.
2 - Contrôle automatique de la conformité des caractères entrés à la base choisie.
3 - Impression automatique des nombres à leur place dans le tableau. Bip sonore.
4 - Si un passage en virgule flottante est nécessaire au cours des calculs, le programme le signale (arrondis possibles).
5 - Programme structuré. Listing clair et commenté.

Bio-rythmes

CARACTERISTIQUES TECHNIQUES:

Fonctionne sur CBM 4000 + EDEX et occupe 6500 octets. une version sans EDEX est disponible.
Enregistré sur cassette.

FONCTION:

Grâce aux bio-rythmes, vous pouvez connaitre votre forme physique, intellectuelle, affective à une date future, présente, passée.

PERFORMANCES ET PRESENTATION:

1 - Affichage de 3 courbes, dont la partie représentée couvre une durée de 1 mois.
2 - Possibilité de mémoriser des prénoms et les dates de naissance correspondantes.
3 - Contrôle des entrées.
4 - Son.
5 - Programme structuré. Listing clair et commenté.

Master-Mind

CARACTERISTIQUES TECHNIQUES:

Fonctionne sur CBM 4000 et occupe 8800 octets.

Enregistré sur cassette.

FONCTION:

Il s'agit du célèbre jeu. L'ordinateur joue le rôle du "codeur".

PERFORMANCES:

1 - Possibilités étendues: jusqu'à 10 symboles dans la combinaison, choix des symboles (chiffres, lettres ou symboles graphiques), jusqu'à 20 symboles différents possibles en jeu, possibilité ou non de répétition des symboles.
2 - Evaluation par l'ordinateur, de la force du jeu, ... et de celle du joueur !
3 - Programme structuré, commenté.

Calendrier an 1 - an 5000

CARACTERISTIQUES TECHNIQUES:

Fonctionne sur CBM 4000 et occupe 5500 octets. Sur cassette.

FONCTION:

Donne, à partir de l'entrée d'un mois et d'un millésime compris entre 1 et 4915, le calendrier correspondant (jour de la semaine et longueur du mois).

DESCRIPTION DU PROGRAMME:

1 - Présentation soignée du calendrier mensuel.
2 - Prise en compte des divers systèmes qui se sont succédé: calendriers julien, grégorien, républicain ...
3 - Il contient donc une partie qui calcule le nombre de jours entre deux dates intéressante.
4 - Programme testé, en particulier sur des dates anciennes.
5 - Contrôle maximal des entrées.
6 - Programme structuré, commenté.

Tracé de courbe

CARACTERISTIQUES TECHNIQUES:

Fonctionne sur CBM 4000 + EDEX et occupe 3800 octets.
Enregistré sur cassette.

FONCTION:

Permet de tracer, à partir du minimum d'entrées la courbe représentative d'une fonction f quelconque, dans la résolution de 4000 points que permet EDEX (Je possède aussi une version sans EDEX).

CAPACITES DU PROGRAMME:

1 - La fonction f est entrée au clavier, et l'instruction DEFFN est engendrée automatiquement.
2 - Le cadrage en abscisses, en ordonnées, le choix des échelles optimales, sont effectués automatiquement.
3 - L'appui sur une touche SHIFT fait apparaître sur les axes les valeurs mini et maxi de x et y; la courbe est ensuite régénérée.
4 - Je prévois la possibilité de pouvoir tracer simultanément plusieurs courbes.
5 - Programme structuré, commenté.

LE FEEDBACK DE LA COMMODE

Pour produire une revue qui vous donne toujours plus de satisfaction, nous avons absolument besoin de vos réponses à ces questionnaires.

Il y a un questionnaire pour chaque numéro. Parmi les réponses concernant chaque numéro. deux seront tirées au sort et leurs auteurs recevront un petit souvenir de La Commode.

Comme au patinage artistique. vous attribuez deux notes à chaque article: une note sur l'intérêt du sujet et une note sur la façon dont le sujet est traité (rédaction. clarté. etc...).Voici les barêmes:

sujet :

intérêt capital................ 4
très intéressant................ 3
assez intéressant.............. 2
pas d'intérêt pour moi......... 1
n'a pas. d'après moi. sa place dans La Commode................ 0

traitement :

excellent.............. 4
bon.................... 3
moyen.................. 2
mauvais................ 1
nul.................... 0

Numéro 7 titre	note sujet	note traitement
Actualité		
Courrier des lecteurs		
Magazine		
La course des compilateurs Basic		
Let there be light ... pen		
Vic à brac		
Création de "sprites" pour CBM 64		
Utilisation des Joystick sur le VIC		
La page du 1er avril		
Les Envahisseurs		
Les astuces de Peter Jollybug		
L'interface RS232 du VIC		
Calcul de factorielle N		
Prêts bancaires		
Le programme TREVERSE		
Ajustement de courbes		
Bibliographie		

Autres commentaires : ....
....
....

Petit questionnaire (réponses indispensables pour être tiré au sort).

nom : .... âge : .....
adresse : ....
....
profession : ....

J'utilise un Commodore modèle ....
pour:
ma profession mes loisirs

Précisez SVP. si vous utilisez différents modèles.

A renvoyer à : La Commode. service FB
28. rue Vicq-d'Azir
75010 PARIS

Pet Club

Il existe à STRASBOURG, au sein du club MICROTEL, une section ouverte au VIC 20. Ce club acceuille tous les systèmes aussi rien n'empêche d'autre CBM-istes à se joindre à nous. L'acceuil se fait le lundi soir de 20h à 22h. La soirée de travail VIC 20 le mardi à partir de 19h30. L'adresse:

87, route du Polygone
67100 STRASBOURG
(à coté de la poste de NEUDORF
au fond de la cour)
tel: (88) 44 11 86

A. MAZET
10 Bd. Gambetta
87000 LIMOGES
cherche dans sa région un club ayant une section VIC 20 et plus généralement Commodore.

Le C.M.I.C.F.S., Club de Micro-Informatique Commodore, Flexette, Sharp,

4, rue Pierre Guys
13012 MARSEILLE

vous aidera à connaître la micro-informatique, la télématique et les télécommunications.

Appel aux clubs

N'hésitez pas à vous faire connaître:

- La Commode tire à 3000 exemplaires.

- Vous rendrez service aux nombreux lecteurs qui cherchent un lieu de rencontre et de dialogue.

Bibliographie

ALGORITHMES et STRUCTURES
de DONNEES,
de R. MAHL et J.C. BOUSSARD
Laboratoire Informatique
Université de Nice
06034 NICE cedex

Ouvrage peu épais (170p) mais d'une extraordinaire densité. Il traite tout d'abord d'une présentation graphique des ordinogrammes très différente de celle habituellement utilisée et à mon avis beaucoup plus claire et efficace. Un chapitre est consacré à la notion importante de temps d'exécution. Les algorithmes de parcours d'arbre sont étudiés avec des exemples très bien choisis pour éclairer les difficiles problèmes rencontrés. Un chapitre excellent sur la récursivité termine la première moitie du livre.

Le reste est consacré à la représentation des informations avec l'étude des suites, des tables, des arbres et des graphes et leurs représentation en mémoire.

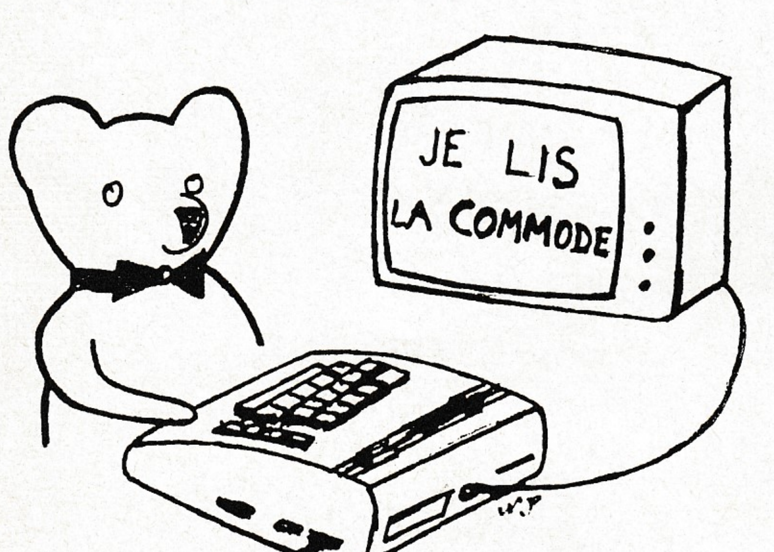
Tous les chapitres son écrits avec clarté et rigueur mais de façon à rester compréhensible même pour les débutants en informatique.

La plupart des solutions sont développées en FORTRAN très facile à traduire en BASIC, ce qui est un exercice fort recommandable.

Parmi la centaine d'ouvrages d'informatique qui me sont passés entre les mains, certainement l'un des meilleurs. Et en français, ce qui ne gâte rien!

Jean DELAVILLE

La Commode 2ᵉ année



Diffusion :
Ed. du P.S.I.
41-51, rue Jacquard
BP 86
77400 LAGNY

Publicité :
FORCE 7
41, rue de la Grange aux Belles
75483 PARIS CEDEX 10
Tél. 238.66.10

Publié par SEDERMI SARL
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Rédaction

Rédacteur en chef :
Daniel TRECOURT

Rédacteur en chef adjoint :
Marc VAN CRAEYNEST

Chefs de rubriques :
Jacques COUTHURES
Jean DELAVILLE
Jean-Luc DESCHAMPS
Pierre-Etienne THALBERG
François VULQUIN

Editeur :
Jean RENAUD

Directeur de la publication :
Daniel-Jean DAVID

Rédaction-vente-abonnements :
28, rue Vicq d'Azir
75010 PARIS
Tél. 205.87.75

Bulletin d'abonnement

A envoyer à La Commode
28, rue Vicq d'Azir - 75010 PARIS

Je désire m'abonner à la Commode (1 an, 4 numéros) à partir du numéro ....

NOM : .... PRÉNOM : ....

Adresse : ....

....

Signature

Règlement 120 F à l'ordre de SEDERMI :

☐ CCP ☐ CB ☐ ESPECES

Réabonnement ☐
Indiquez si possible votre ancien numéro d'abonné

Imprimerie Laballery, 58500 Clamecy. Imprimé en France. Directeur de la publication Daniel-Jean David
Dépôt légal : mai 1983 - Numéro de commission paritaire : en cours ISSN 0292-5079



VIC 20: L'ORDINATEUR-COPAIN.

Avec le Commodore Vic 20 vous vous faites 4 copains d'un seul coup.

Un copain pour le jeu; Vic 20 est champion : Échecs, Guerre des Étoiles, Alien, Les Envahisseurs..., vous passerez des heures avec lui.

Un copain pour les études; Vic 20 vous apprend les maths ou les statistiques ou l'anglais..., en vous amusant.

Un copain pour l'initiation à l'informatique; pas de problème, un cours pratique d'autoformation vous apprendra le langage informatique. Vous pourrez créer ainsi vos propres programmes.

Un copain pour la gestion familiale; avec ses programmes de comptabilité, de gestion de fichiers et calculs électroniques, Vic 20 aidera toute la famille.

Il y a plus de 200 revendeurs Commodore en France. Venez les voir. Pour Noël, ils vous réservent une bonne surprise à un prix tout à fait amical.

COMMODORE VIC 20
LE COPAIN DE TOUTE LA FAMILLE.

Pour 3200 F* TTC seulement:
— Le Vic 20 avec interface noir et blanc intégré — Le lecteur de cassette — Le cours d'autoformation au Basic comprenant un manuel d'utilisation et 2 cassettes de travaux pratiques.
— Une cassette comprenant deux jeux: Blitz et Night Park.
— Une cassette comprenant deux programmes:
• 1 programme éducatif: Calcul Spatial;
• 1 programme gestion: Budget Familial.

*Offre spéciale valable jusqu'au 31/12/82.

CONCURRENCE MODERNE